

THESIS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

# Development of fine-mesh methodologies for coupled calculations in Light Water Reactors

KLAS JARETEG



Department of Physics  
CHALMERS UNIVERSITY OF TECHNOLOGY  
Göteborg, Sweden 2017

Development of fine-mesh methodologies for coupled calculations in Light Water Reactors  
KLAS JARETEG  
ISBN 978-91-7597-626-6

©Klas Jareteg, 2017

Doktorsavhandling vid Chalmers tekniska högskola  
Ny serie nr 4307  
ISSN 0346-718X

Division of Subatomic and Plasma Physics  
Department of Physics  
Chalmers University of Technology  
S-412 96 Göteborg  
Sweden  
Telephone +46 (0)31 772 1000

Cover: Scalar neutron flux at mid-elevation for the thermal group in a steady-state coupled simulation (described in Section 4.2).

Typeset using  $\text{\LaTeX}$  and TikZ

Chalmers Reproservice  
Göteborg, Sweden 2017

Development of fine-mesh methodologies for coupled calculations in Light Water Reactors  
KLAS JARETEG  
Division of Subatomic and Plasma Physics  
Department of Physics  
Chalmers University of Technology

## **ABSTRACT**

This thesis presents fine-mesh multiphysics methodologies and algorithms for numerical predictions of the behavior of Light Water Reactor (LWR) cores. The multiphysics aspects cover the distribution of neutrons, the fluid flow of the coolant and the conjugate heat transfer between the solid fuel pins and the fluid coolant. The proposed schemes are aimed at fine-mesh coupled effects, directly resolving the interdependencies of the different fields on the finest scales of the computations.

The solver is developed for both steady-state and transient LWR scenarios. For the steady-state simulations, the neutronics is solved both by the lower order, diffusion equation and the higher order, discrete ordinate transport method, and for transient cases by the former. The thermal-hydraulic solver is based on a computational fluid dynamics (CFD) approach. The implementation utilizes a finite volume method (FVM) computational framework, and to achieve feasible computational times, high performance computing (HPC) aspects such as parallelization by domain decomposition are considered.

The implemented tool is applied to cases of parts of a fuel assembly, analyzing systems of up to  $15 \times 15$  fuel pins and successfully resolving sub-pin resolution of all fields. Furthermore, the transient fine-mesh neutronic solver is verified based on a novel scheme utilizing the system response to a local perturbation.

In addition, the multiphase flow problem encountered in Boiling Water Reactors (BWRs) is studied. First, the transport of bubbles under subcooled boiling conditions is simulated based on a population balance approach. The novel formulation is shown to increase the computational efficiency and to capture a large range of bubbles sizes with few degrees of freedom. Second, the typical Eulerian-Eulerian approach for two-phase flow is studied from a stability and dynamics perspective. The latter investigations highlight the complexity of the two-fluid formulation and indicate the spontaneous emergence of meso-scale void structures under adiabatic conditions.

*Keywords:* Coupled neutronics/thermal-hydraulics, CFD, nuclear reactor multiphysics, multiphase flow



## Preface

The process of writing and defending a thesis is a long and interesting one, not least so in the politically and socially complicated field of nuclear power. The Swedish nuclear industry and research have both seen drastic reverses during my few years in the field and the future is, for better and worse reasons, indecisive. As a PhD student it has however been my privilege to focus primarily on the technological contributions. It is my sincere hope that the work presented in this thesis will help to improve the field of nuclear engineering.

It has been my intention to write a thesis which could be read not only by experts in nuclear reactor simulations but also by a more general audience. In particular the introduction is written to give an overview of some current simulation schemes, their shortcomings and some of the late developments in HPC multiphysics simulations for reactors. However, to keep this summary at a reasonable length I can only allow myself to scratch the surface.

The second chapter (Computational Methodology) is meant both as an introduction to methods relevant for the current work as well as to explain and discuss some of the choices that were done during the code development and setup of the multiphysics framework. My hope is that this chapter is clear and detailed enough as to allow someone else starting a similar project to understand and actually learn some of the practical details that I spent (a lot of) time on.

The thesis makes an attempt to tackle the multiphysics problem of the reactor core. As such, my research is at best multifaceted and at worst scattered. Nevertheless, it has been my intention to provide a holistic description of the methodologies developed and applied in this thesis. Consequently, a significant proportion of the details of the results of the applications are left for the interested reader to find in the appended papers.

I am grateful towards a large number of people (as further expressed in the Acknowledgements), and not least to my three supervisors for assisting and contributing to this work. All remaining errors are my own.

Klas Jareteg  
Gothenburg 2017/06/27



## Appended papers

This thesis is an introduction to and a summary of the work published in the following papers

- PAPER I      **K. Jareteg**, P. Vinai, C. Demazière. “Fine-mesh deterministic modeling of PWR fuel assemblies: Proof-of-principle of coupled neutronic/thermalhydraulic calculations”. *Annals of Nuclear Energy* 68 (2014), pp. 247 –256
- PAPER II     **K. Jareteg**, P. Vinai, S. Sasic, C. Demazière. “Influence of an  $S_N$  solver in a fine-mesh neutronics/thermal-hydraulics framework”. *PHYSOR 2014, September 28 - October 3, Japan* (2014)
- PAPER III    H. Ström, S. Sasic, **K. Jareteg**, C. Demazière. “Behaviour and Stability of the Two-Fluid Model for Fine-Scale Simulations of Bubbly Flow in Nuclear Reactors”. *International Journal of Chemical Reactor Engineering* 13.4 (2015), pp. 449–459
- PAPER IV    **K. Jareteg**, P. Vinai, S. Sasic, C. Demazière. “Coupled fine-mesh neutronics and thermal-hydraulics - modeling and implementation for PWR fuel assemblies”. *Annals of Nuclear Energy* 84 (2015), pp. 244 –257
- PAPER V     **K. Jareteg**, R. Andersson, C. Demazière. “Development and test of a transient fine-mesh LWR multiphysics solver in a CFD framework”. *M&C 2015, Nashville, Tennessee* (2015)
- PAPER VI    **K. Jareteg**, S. Sasic, P. Vinai, C. Demazire. “A numerical framework for bubble transport in a subcooled fluid flow”. *Journal of Computational Physics* 345 (2017), pp. 373–403
- PAPER VII   **K. Jareteg**, H. Ström, S. Sasic, C. Demazière. “On the dynamics of instabilities in two-fluid models for bubbly flows”. *Chemical Engineering Science* 170 (2017), pp. 184–194
- PAPER VIII   C. Demazière, V. Dykin, **K. Jareteg**. “Development of a point-kinetic verification scheme for nuclear reactor applications”. *Journal of Computational Physics* 339 (2017), pp. 396–411

## Contribution report

PAPER I	KJ formulated the model, implemented the code, performed all simulations and wrote the paper together with the coauthors.
PAPER II	KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.
PAPER III	KJ contributed to the analysis, ideas for simulations and the preparation of the final version of the paper.
PAPER IV	KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.
PAPER V	KJ formulated the model and implemented the code together with RA, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.
PAPER VI	KJ formulated the model, implemented the code, performed all simulations and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.
PAPER VII	KJ formulated parts of the model, performed all simulations, contributed to the analysis and wrote the first draft of the paper. The final version of the paper was prepared with support from the coauthors.
PAPER VIII	KJ contributed with the implementation of the time-dependent version of the code and the corresponding verification simulations and data analysis, and was involved in the writing of the corresponding sections in the paper.



## Other related publications, not included in the thesis

**K. Jareteg**, P. Vinai, C. Demazière. "Investigation of the possibility to use a fine-mesh solver for resolving coupled neutronics and thermal-hydraulics". *M&C 2013, Sun Valley, Idaho* (2013)

H. Ström, S. Sasic, **K. Jareteg**, C. Demazière. "On the validity of the two-fluid model for simulations of bubbly flow in nuclear reactors". *Proceedings of the 13th International Conference on Multiphase Flow in Industrial Plants (MFIP-13)*. 2014

C. Demazière, **K. Jareteg**. "Developing a course in nuclear reactor modelling and going from campus-based to web-based teaching". *PHYSOR 2014, September 28 - October 3, Japan* (2014)

C. Demazière, **K. Jareteg**. "Students and teachers perspectives in going from campus-based to web-based teaching". *Conference on Teaching and Learning (Konferens om Undervisning och Lrande KUL2014)*. 2014

H. Ström, **K. Jareteg**, S. Sasic, C. Demazière. "Two-fluid model analyses of instabilities and non-uniformities in bubbly gas-liquid flows". *12th International Conference on Gas-Liquid and Gas-Liquid-Solid Reactor Engineering (GLS12)*. 2015

**K. Jareteg**, S. Sasic, P. Vinai, C. Demazière. "Development of a Coupled Two-Fluid/DQMOM Methodology for Heated Bubbly Flows". *CMFF2015, Budapest, Hungary* (2015)

**K. Jareteg**, H. Ström, S. Sasic, C. Demazière. "Numerical investigation of instabilities in the two-fluid model for CFD simulations of LWRs". *M&C 2015, Nashville, Tennessee* (2015)

A. Jareteg, **K. Jareteg**, S. Sasic. "Formulation of stresses in dry granular flows". *CMFF2015, Budapest, Hungary* (2015)

E. Pettersen, C. Demazière, **K. Jareteg**, E. Schönfeldt T, B. Lauritzen. "Development of a Monte-Carlo based method for calculating the effect of stationary fluctuations". *M&C 2015, Nashville, Tennessee* (2015)

K. Subramani Kannan, **K. Jareteg**, N. Lassen, J. Carstensen, M. Edberg Hansen, F. Dam, S. Sasic. "Design and performance optimization of gravity tables using a combined CFD-DEM framework". *Powder Technology* 318 (2017), pp. 423 –440

C. Demazière, V. Dykin, **K. Jareteg**. "Development and Test of a New Verification Scheme for Transient Core Simulators". *ANS Annual Meeting June 11-15, San Fransisco, 2017*. 2017

G. Sardina, **K. Jareteg**, H. Ström, S. Sasic, C. Demazière. “The nature of instabilities in bubbly flows / a comparison between Eulerian-Eulerian and Eulerian-Lagrangian approaches”. *MFIP2017, Desenzano del Garda, 13-15 September 2017* (2017)

# Abbreviations & Nomenclature

## Abbreviations

<b>BWR</b>	Boiling Water Reactor
<b>CFD</b>	Computational Fluid Dynamics
<b>CHF</b>	Critical Heat Flux
<b>CHT</b>	Conjugate Heat Transfer
<b>CRUD</b>	Chalk River Unidentified Deposits
<b>CV</b>	Control volume
<b>DNS</b>	Direct Numerical Simulation
<b>DQMOM</b>	Direct Quadrature Method of Moments
<b>FDM</b>	Finite Difference Method
<b>FEM</b>	Finite Element Method
<b>FVM</b>	Finite Volume Method
<b>HPC</b>	High performance computing
<b>LES</b>	Large Eddy Simulation
<b>LWR</b>	Light Water Reactor
<b>MUSIG</b>	Multiple-size-group method
<b>NK</b>	Neutron kinetics
<b>NPP</b>	Nuclear Power Plant
<b>PBE</b>	Population balance equation
<b>PDE</b>	Partial differential equation
<b>PISO</b>	Pressure Implicit with Split Operator
<b>PWR</b>	Pressurized Water Reactor
<b>RANS</b>	Reynolds Averaged Navier-Stokes equations
<b>SIMPLE</b>	Semi-Implicit Method for Pressure Linked Equations
<b><math>S_N</math></b>	Discrete ordinates method
<b>TH</b>	Thermal-hydraulics

## Nomenclature

### General

$\delta_{ij}$	Kroenecker delta
$\rho$	Density
$\Theta$	Example quantity
$\Theta_p$	$\Theta$ for cell $p$
$\Theta_s$	Face interpolated value of $\Theta$
$\Omega$	Angular direction
$g$	Gravitational acceleration
$\mathbf{r}$	General space coordinate
$S_f$	Face area vector
$S_\Theta$	Source term for $\Theta$
$t$	General time coordinate
$V$	Volume of mesh cell

### Neutronics

$\beta$	Fraction of delayed neutrons
$\gamma$	Energy per fission
$\lambda$	Decay constant
$\mu$	Average scattering angle
$\nu$	Neutron fission yield
$\rho_A$	Atomic density
$\sigma_x$	Microscopic cross-section for reaction $x$
$\Sigma_a$	Absorption cross-section
$\Sigma_f$	Macroscopic fission cross-section
$\Sigma_s$	Macroscopic scattering cross-section
$\Sigma_{s0}$	Macroscopic isotropic scattering cross-section
$\Sigma_T$	Total macroscopic cross-section
$\phi$	Scalar neutron flux
$\varphi$	Expansion coefficient in real spherical harmonics base
$\chi$	Fission neutron spectrum
$\Psi$	Angular neutron flux
$C$	Precursor concentration
$D$	Diffusion coefficient
$F$	Fission source term
$G$	Number of energy groups
$\mathbf{J}$	Neutron current
$k_{\text{eff}}$	Multiplication factor
$n$	Neutron density
$P_N$	Power density
$P_l$	Legendre polynomials
$R_{lr}$	Real spherical harmonics
$S$	Scattering source
$w$	Quadrature weight

### TH - Single-phase

$\alpha$	Isothermal compressibility coefficient
$\beta$	Thermal expansion coefficient
$\epsilon$	Dissipation of turbulent kinetic energy
$\mu$	Dynamic viscosity
$\mu_t$	Turbulent kinetic viscosity
$\tau$	Stress tensor
$c_p$	Specific heat capacity at constant pressure
$h$	Instantaneous enthalpy
$H$	Time averaged enthalpy
$k$	Turbulent kinetic energy
$K$	Thermal conductivity
$p$	Instantaneous pressure
$P$	Time averaged pressure
$q''$	Surface heat flux
$q'''$	Volumetric heat source
$T$	Temperature
$u$	Instantaneous velocity
$\mathbf{U}$	Time averaged velocity

### TH - multiphase

$\alpha$	Void fraction
$\xi$	Abscissa (bubble size)
$\mu$	Dynamic viscosity
$\rho$	Density
$\bar{\tau}$	Stress tensor
$\bar{\tau}^t$	Turbulent stress tensor
$\Phi$	Time resolved uniformity index
$C$	Condensation rate
$d_{43}$	Mean diameter
$d_b$	Bubble size
$f$	Average number density
$g$	Vapour phase
$i$	Phase
$j$	Bubble size index
$l$	Liquid phase
$\mathbf{M}$	Momentum transfer due interfacial forces
$N$	Number of abscissas
$P$	Pressure
$S$	Source term for condensation, aggregation and breakage
$t$	Turbulent quantity
$\mathbf{U}$	Phase velocity
$w$	Weight

# Contents

<b>Abstract</b>	<b>I</b>
<b>Preface</b>	<b>III</b>
<b>Appended papers</b>	<b>V</b>
<b>Contribution report</b>	<b>VI</b>
<b>Other related publications, not included in the thesis</b>	<b>VII</b>
<b>Abbreviations &amp; Nomenclature</b>	<b>IX</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Multiphysics in the reactor core . . . . .	1
1.2 Neutronics and thermal-hydraulic simulations . . . . .	5
1.2.1 Lattice and core simulations of the neutronics . . . . .	5
1.2.2 System codes for thermal-hydraulics . . . . .	7
1.2.3 Segregated multiphysics schemes . . . . .	9
1.3 High-fidelity core simulations . . . . .	9
1.3.1 Motivations for novel approaches . . . . .	9
1.3.2 Overview of multiscale and multiphysics approaches . . . .	10
1.4 Objectives of the research work . . . . .	11
1.4.1 Fine-mesh multiphysics simulation objectives . . . . .	12
1.4.2 Multiphase flow objectives . . . . .	13
1.4.3 Outline of the thesis . . . . .	15
<b>2 Computational methods</b>	<b>17</b>
2.1 High Performance Computing . . . . .	18
2.1.1 A brief historical perspective . . . . .	18
2.1.2 Code efficiency and optimization . . . . .	19
2.1.3 Parallelization . . . . .	20
2.2 The finite volume method . . . . .	23
2.2.1 Equation discretization . . . . .	23
	<b>XI</b>

2.2.2	Sparse matrix solvers . . . . .	25
2.2.3	OpenFOAM® . . . . .	26
2.3	Computational scheme . . . . .	28
2.3.1	Grid generation . . . . .	29
2.3.2	Mesh mapping . . . . .	35
2.3.3	Cross-section generation and utilization . . . . .	36
<b>3</b>	<b>Models for the coupled LWR single-phase problem</b>	<b>41</b>
3.1	Formulation of the neutronic problem . . . . .	41
3.1.1	Diffusion approximation . . . . .	44
3.1.2	Discrete ordinates method . . . . .	45
3.2	Single-phase fluid flow and heat transfer . . . . .	47
3.2.1	Turbulence . . . . .	48
3.2.2	Pressure and velocity algorithms . . . . .	49
3.2.3	Heat transfer problem . . . . .	50
3.3	Multiphysics formulation and algorithms . . . . .	51
<b>4</b>	<b>Steady-state coupled solver application and analysis</b>	<b>55</b>
4.1	Implementation and framework details . . . . .	55
4.1.1	SIMPLE algorithm and heat transfer . . . . .	56
4.1.2	Discrete ordinates solver . . . . .	57
4.1.3	Picard iteration scheme . . . . .	58
4.2	Application to a $15 \times 15$ assembly . . . . .	59
4.2.1	Convergence and performance . . . . .	60
<b>5</b>	<b>Transient coupled solver application and analysis</b>	<b>65</b>
5.1	Implementation and framework details . . . . .	65
5.2	Application to a $7 \times 7$ assembly . . . . .	66
5.3	Time-dependent neutronics verification method . . . . .	68
5.3.1	Overview of the methodology . . . . .	70
5.3.2	Application to a two-region slab system . . . . .	71
<b>6</b>	<b>On two-phase flow in LWRs</b>	<b>73</b>
6.1	Perspectives on the simulation challenges . . . . .	74
6.2	Models for two-phase liquid and gas flows . . . . .	76
6.2.1	DNS-like methods . . . . .	76
6.2.2	The two-fluid model . . . . .	77
6.3	Population balance for subcooled bubbly flow . . . . .	79
6.3.1	PBE formulation . . . . .	80
6.3.2	Application to channel flow with condensation . . . . .	81
6.4	On the dynamics of the two-fluid formulation . . . . .	84
6.4.1	Application to adiabatic cases . . . . .	85

<b>7</b>	<b>Conclusions and recommendations for future work</b>	<b>89</b>
7.1	Fine-mesh multiphysics simulations . . . . .	89
7.1.1	Summary of the methodology . . . . .	89
7.1.2	Results and conclusions . . . . .	90
7.2	Two-fluid simulations . . . . .	91
7.2.1	DQMOM coupled to a two-fluid solver . . . . .	91
7.2.2	Two-fluid instability results . . . . .	92
7.3	Future outlook . . . . .	93
	<b>Acknowledgements</b>	<b>95</b>
	<b>References</b>	<b>97</b>
	<b>Papers I-VIII</b>	<b>109</b>
	Papers I . . . . .	111
	Papers II . . . . .	123
	Papers III . . . . .	143
	Papers IV . . . . .	157
	Papers V . . . . .	173
	Papers VI . . . . .	191
	Papers VII . . . . .	225
	Papers VIII . . . . .	239





# Introduction

This thesis begins with, and ultimately originates from, the physics in the core of the nuclear power plant (NPP). The core is at the heart of a reactor in every aspect, located in the center of the plant and containing the fuel with its enormous potential of heat generation from fissioning of heavy nuclei. The reactor core is a complex environment, governed by multiple fields of intertwined and coupled physics influencing the process on a wide range of length scales. This environment is the field of this thesis and the main goal shall be to increase the understanding of how such a system can be simulated with high resolution and accuracy.

This first chapter will introduce the main fields of physics that are of interest in the reactor core and in particular how these are mutually coupled (Section 1.1). To give a context of the contribution of this work, some standard (or even classical) schemes of nuclear reactor core simulations (Section 1.2) are described, followed by some more recent developments in reactor modeling (Section 1.3). Finally, the objectives of this thesis are presented in detail in Section 1.4.

## 1.1 Multiphysics in the reactor core

The core of the Light Water Reactor (LWR) contains the solid uranium fuel pins which contribute with the heat source in the reactor. A controlled chain reaction of fissions results in a continuous and enormous release of energy in the reactor core. The energy is conducted via the solid encapsulation, the cladding, of the fuel and then extracted from the pins and convected out of the core via the heated water. To understand the interplay between the neutron density and the fluid flow and heat transfer in the coolant, a more detailed description of the multiphysics is of interest.

The chain reaction of fissions is governed by the distribution of neutrons in

the core. If the population of neutrons is kept at a statistically steady concentration, the reactor is said to be critical. On the other hand, if there is an increase in neutrons with time, the reactor is in a supercritical state. Reversely, if fewer neutrons are born from fission than disappear due to fission, absorption or leakage, the reactor is in a subcritical state. To achieve an economic utilization of the fuel, the core should be designed to minimize the loss of neutrons due to leakage out of the physical domain and due to other nuclear reactions.

The cross-section for a certain reaction, i.e. the probability for a neutron to interact with a certain target, is determined by properties of the interacting material as well as the energy of the neutron. The microscopic cross-section ( $\sigma$ ) describes the probability for a certain reaction to take place, such as absorption, capture or fission to occur (see details below). Due to the dependence on the density of the target material, the neutron distribution is influenced by all processes which result in variations of the material concentrations. For this thesis, the coupling to the water density is of particular interest. In addition, the temperature of the fuel has a significant effect on the cross-sections due to the so-called Doppler broadening.

#### Microscopic and macroscopic cross-sections

The microscopic cross-section gives the probability for a certain reaction to occur given the incident neutron energy such that the reaction rate  $R_x$  is given by

$$R_x(E) = \sigma_x(E)\phi(E)\rho_A, \quad (1.1)$$

where  $\phi(E)$  denotes the neutron flux and  $\rho_A$  the atomic density of the target material. Commonly the combination of the microscopic cross-section and the density is written as a macroscopic cross-section

$$\Sigma_x(E) = \sigma_x(E)\rho_A. \quad (1.2)$$

The figure below shows the microscopic fission cross-section of Uranium-235. For low energies the cross-section is inversely proportional to the neutron velocity, whereas a much more complicated dependence is seen for the intermediate region. The resonances in this energy range constitute a particularly challenging task in the process of cross-section condensation as an average of a longer energy interval in the resonance region strongly depends on the precise spectrum of the neutron flux.

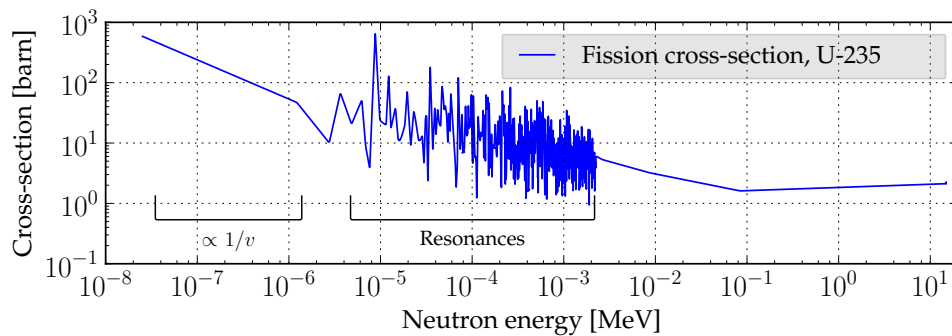
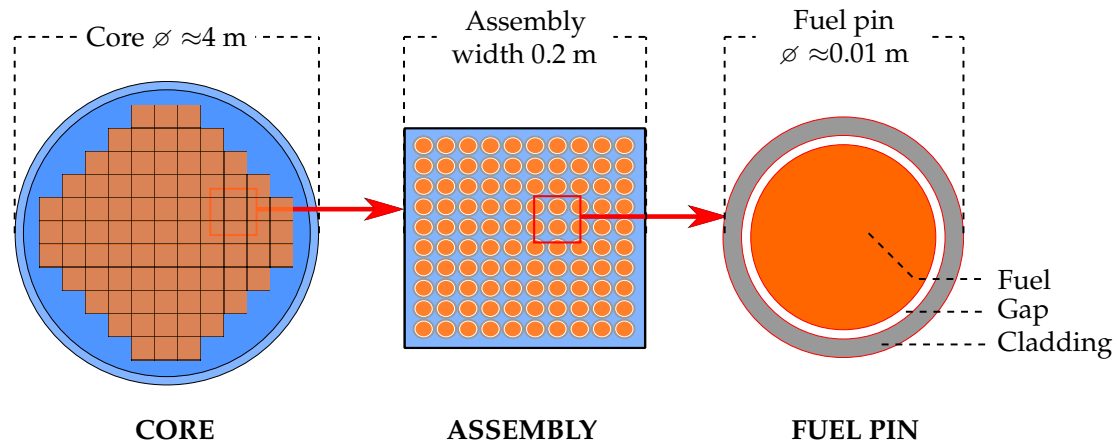


Figure 1.1: Fission cross-section for Uranium-235 [1].



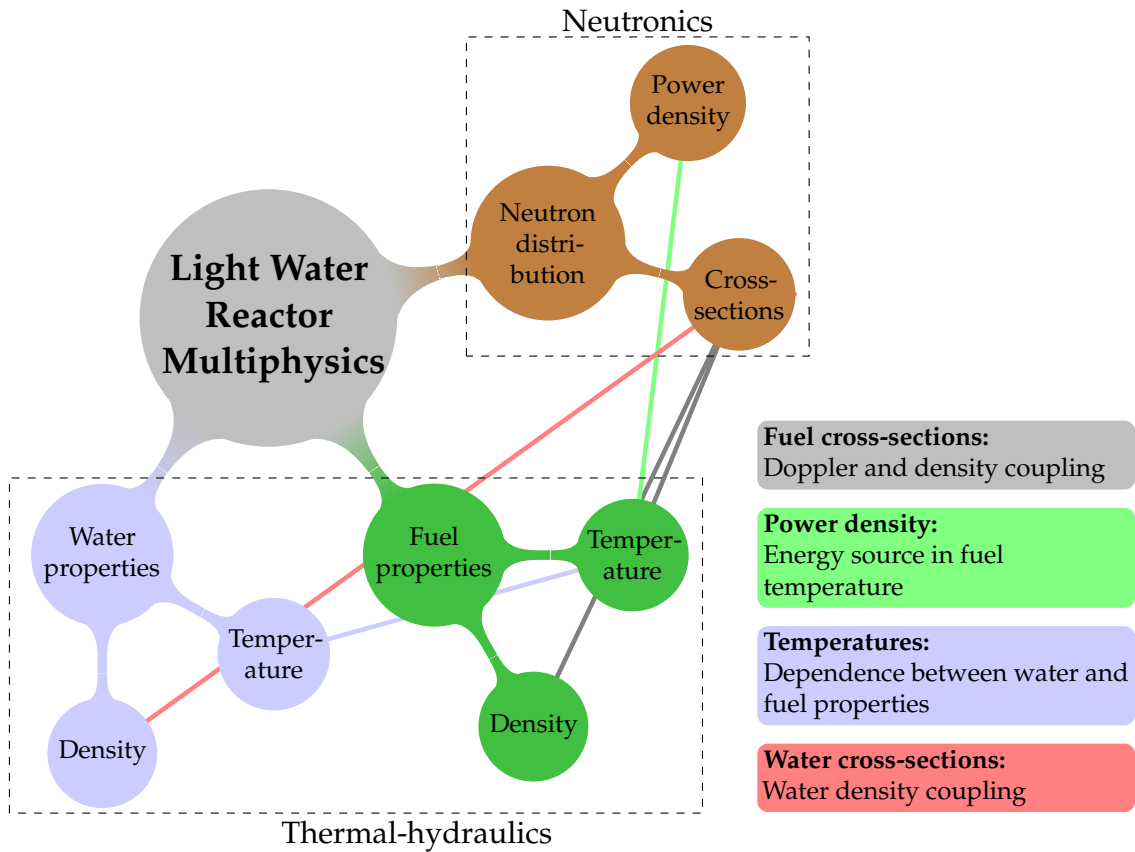
**Figure 1.2:** Schematic drawing of a horizontal plane of a LWR core.

The temperature profile in the fuel is determined by the energy release from the fissions and the conduction in the fuel pins, i.e. the transport of energy in the stack of fuel pellets in each rod in the assembly (see Figure 1.2). The conductivity of the solid fuel matrix is governed by the micro structure as well as the temperature of the material. Due to the neutron irradiation in the core, the fuel and cladding properties change with time in the reactor. The processes responsible for such defects constitute a complete field of material science, and complex phenomena such as cracking of the fuel might take place.

To allow thermal expansion and fuel swelling, LWR fuel rods have a small gap between the fuel and the cladding, initially filled by an inert gas. Due to the small distance between the solids, the major mechanism for heat transfer (under normal reactor operating conditions) is conduction also in the gap. The cladding is the first safety barrier in the reactor, designed to contain the fuel and the fission products. Although no fission events occur in the gap or cladding, these regions are still influential on the neutronics problem as the neutron distribution is affected via capture.

The heat conducted from the cladding is extracted from the core with forced convection. In the case of the LWR, the water acts as the coolant, and a high flow velocity through the core is maintained by pumps. In the case of a Pressurized Water Reactor (PWR) the water is nominally kept in liquid state, whereas a phase change from liquid to vapor is seen in a Boiling Water Reactor (BWR). The high flow velocity results in a turbulent flow with enhanced heat transfer properties. The turbulence in the coolant is further enhanced by the so-called spacers, essentially steel frames holding the fuel pins as well as interrupting the flow and inducing swirls.

In addition to acting as a coolant, the water in a LWR functions as a moderator for the neutrons. The moderation process cools, i.e. slows down, the high energy neutrons born from fission. The benefit of a higher concentration of low energy neutrons is best understood from the energy dependence of the fission



**Figure 1.3:** Diagram of the multiphysics couplings for the neutronics and thermal-hydraulics in the LWR core.

cross-section for U-235 as presented in Figure 1.1. Due to the inverse proportion of the microscopic fission cross-section, the chance for fission events is much larger for slow neutrons. The moderation primarily occurs due to elastic scattering between the neutrons and the hydrogen atoms in the water.

All described physical processes are connected and the reactor core problem is thus a true multiphysics problem in the sense that one field cannot be solved without knowledge of the others. To summarize the different couplings, Figure 1.3 shows the primary dependencies between the neutron behaviour, the so called neutronics, and the fluid flow and heat transfer in the water and the fuel, the so called thermal-hydraulics. The temperature and density of the fuel and the water both influence the neutronics, which in turns affects the fuel temperature directly through the energy release from fission, and water indirectly as the heat is transported from the fuel to the water.

In addition to the multiphysics aspects of the reactor core, the problem is further complicated by the many length scales to be resolved. The schematic representation in Figure 1.2 shows a horizontal plane in the core. The figure outlines a hierarchy of relevant scales, ranging from the full core size, via the fuel assem-

blies, to the separate fuel pins. In principle, the hierarchy could be continued with even smaller scales governing some of the phenomena discussed above (e.g. turbulence in the water and micro structures of the fuel matrix). In the extreme, we could even consider the atomic length scales and the sizes of the nucleons which ultimately govern the interaction between the neutrons and the nuclei.

Due to the extreme range of scales, it is immediately clear that it is not possible to directly resolve all parts of the problem from first principles. In order to solve the ultimate problem of the complete reactor core, we must rely on assumptions and closures from other scales. Inevitably, such closures introduce errors. As the reactor is also a multiphysics environment, the use and derivation of closures for the large scale problems must also consider that coupled physics phenomena occurring at the small scales need to be correctly represented at the larger scales, which is far from trivially granted. The latter is a key aspect to why the focus in this thesis is on fine-mesh simulations, i.e. simulations where the different fields of physics can be directly coupled without the approximations required to solve the full core problem.

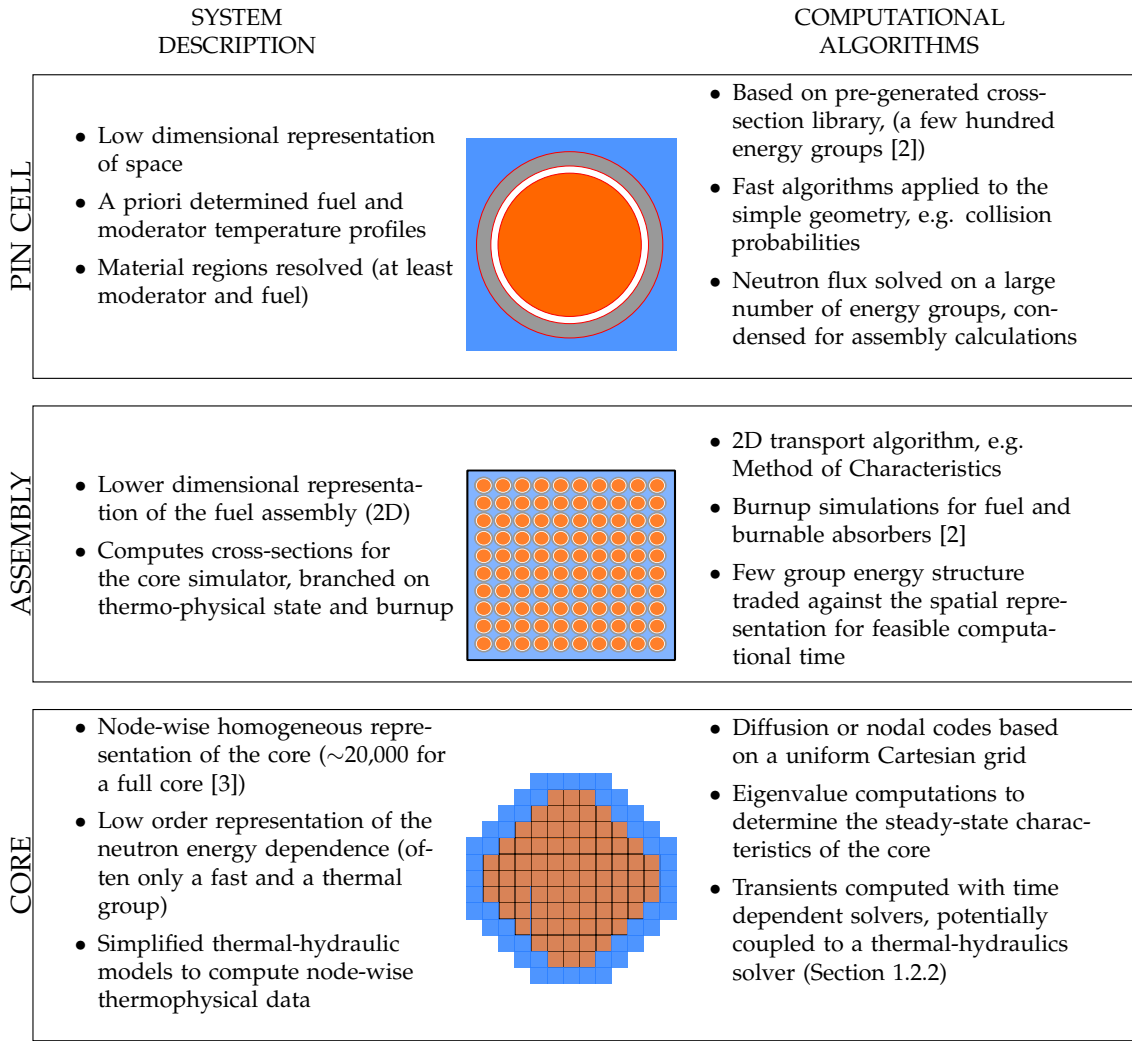
## 1.2 Neutronics and thermal-hydraulic simulations

Given the multiscale and multiphysics problem of the reactor core, we now turn the attention to different options and strategies to simulate and produce numerical predictions of the behavior of the core. To motivate the need of novel algorithms, a brief overview is first presented of some standard schemes applied in routine calculations for the core. The described procedures are well established and the current practices have prevailed for many decades in the same or at least similar shapes. Nevertheless, there are significant limitations and assumptions for multiphysics perspectives and for the physics of the finest scales resolved.

### 1.2.1 Lattice and core simulations of the neutronics

Although a rapid increase is seen in the use of Monte Carlo methods for the neutronic calculations, even on a full core scale, such methods are still too computationally expensive for routine industrial calculations (further discussed in Section 1.3). As a result, the industry still relies on deterministic computations for the neutronic problem, and the predominant schemes have for a long time relied on hierarchal algorithms. Such multiscale approaches range from the simulations of a single pin cell (i.e. a fuel pin surrounded by the coolant), to fuel assembly calculations and finally to the full core scale. A brief overview of the hierarchy of scales is shown in Figure 1.4, where the three different levels are outlined together with some standard choices of algorithms.

In detail, the first two stages of the simulations (pin cell and assembly calculations) are typically computed in a lattice code. Conveniently, such a code



**Figure 1.4:** Overview of a multiscale deterministic neutronics scheme progressing from pin-wise 1D calculations, via 2D assembly calculations to full core 3D simulations

generates a library of cross-sections which are branched to cover desired state points for fuel burnout, thermo-physical state of the reactor, control rod positions etc. Due to the large number of different fuel assemblies and feasible states, the lattice calculators must rely on fast algorithms and sufficiently coarse approximations to give useful simulation times.

Relevant for this thesis, the approximations in the lattice code include assumptions of approximate fuel and moderator temperature profiles. Whereas the actual horizontal temperature profiles are multiphysics dependent (as depicted in Figure 1.3), a standard lattice code is run with explicit, and often discrete, temperature profiles.

The last stage, the core calculation, is for LWRs performed on a coarse Cartesian grid corresponding to the fuel assemblies in the core. A large number of

solver variations exist. However, the core simulators again require fast computations as the codes are routinely applied repetitively e.g. in the process of core design.

Furthermore, the core solver relies on lower dimensional thermal-hydraulic algorithms to predict the temperature and void distributions in the core. The computed states are used to interpolate the cross-section tables from the lattice solver. Roughly, each fuel assembly has one associated thermal-hydraulic channel in which the conservation equations of mass, momentum and energy are solved (see e.g. [4]). At the scale of the core solver, the multiphysics couplings are coarse and by no means resolve small fluctuations or pin-wise temperature profiles.

From this crude description of the neutronic scheme it is clear that the methodology is streamlined to give fast computations for the full core problem with the core solver and the cross-section tables computed in the lattice solver. The multiphysics aspects are limited and the direct coupling to the (simplified) thermal-hydraulic solvers is performed only at the coarsest level.

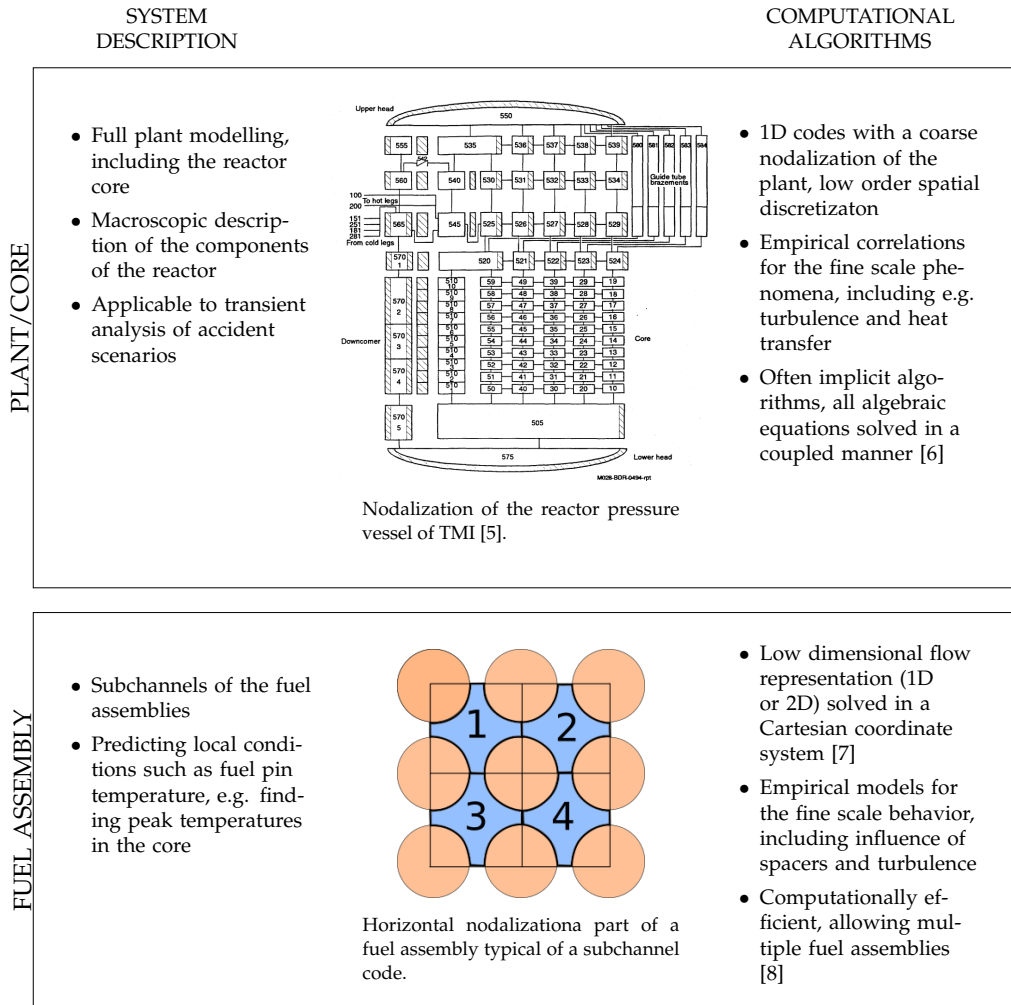
### 1.2.2 System codes for thermal-hydraulics

The thermal-hydraulic counterpart to the hierarchal neutronics methodology consists of a full core solver, partitioned on so-called channels, and a sub-channel code simulating one or a few such channels with a higher resolution. Figure 1.5 presents a brief overview and characteristic algorithms for both methodologies. In contrast to the neutronics scheme, the thermal-hydraulic solvers are not necessarily combined in the same workflow.

The goal of the system code is to compute the complete plant response to a set of transient scenarios, including accidents such as a loss-of-coolant accident [9]. The representation of the flow is much simplified, relying on 1D transport equations, discretized with first order schemes in space and time [6]. As illustrated in Figure 1.5, the core, as well as the other components, is treated with a coarse nodalization. As a result of the crude representation, the effects from for example fluid fluctuations are not modeled but included in correlations. The benefit of the coarse approach is the short wall clock time for relatively long transient scenarios, even on desktop computers.

The algorithms for the neutronic response are typically based on much simplified models, such as the 0D point-kinetic model [10]. On the other hand, for scenarios where the neutronic response is crucial, a coupling to a 3D neutronic solver is advantageous (which is further discussed in Section 1.2.3).

In partial contrast to the macroscopic full plant modeling in the system code, a subchannel code is aimed at (more) local conditions in specific fuel assemblies. For safety reasons it is of interest to e.g. find the hottest channels in the reactor core in order to determine the peak cladding temperature (PCT) [11]. For such a



**Figure 1.5:** Overview of the characteristics of a coarse full plant system code and a higher resolution subchannel code.

purpose the subchannel approach gives more information than the coarse nodalization in the system code. From fluid dynamics and heat transfer perspectives, the subchannel code is still very far from the first principles, relying on correlations for pressure drops, heat transfer, multiphase flow and more [12]. Furthermore the flow is still not resolved in 3D manner but only 2D (see e.g. [7]).

In addition to the mentioned schemes, there is of course the option of a further resolved and much more fine grained simulation. In nuclear terms this would be denoted as CFD, although in principle also the previously described models are numerical predictions of fluid flows, thus deserving the epithet CFD. Although there are numerous examples of interesting core related applications of CFD in the open literature, the 30 years of experience of the macroscopic type of modelling should not be forgotten [13]. The current and future potentials of CFD are further discussed below (Section 1.3).



### 1.2.3 Segregated multiphysics schemes

For some accident scenarios the multiphysics aspects need to be better resolved than it is in the case of the system codes with simplified neutronic models [10]. For such scenarios, the classical choice is to perform an external code coupling between a system code and a neutronic core solver. Even though appreciated as multiphysics in a more integrated sense, the coupling algorithms are commonly based on the so-called operator splitting techniques [14] with low order time schemes. Nevertheless, such best estimate approaches are a key tool for current industrial simulation routines, in particular due to the enormous efforts of code verification and validation already spent on this type of simulations for industrial plants. Examples of attempts on higher order time schemes exist, e.g. by implicit formulations of a combined system matrix [15], such and similar attempts are still limited in the resolved scales of multiphysics. The mapping between the codes is often coarse in the sense that only macroscopic quantities or average properties are exchanged. Consequently, there is no increased fidelity in the physics simulated. The coupled calculations are limited by the very static geometries and resolutions of each of the separate codes and the simplistic coupling schemes and mappings. In contrast, the type of projects next described (Section 1.3) are to a majority focused on newly developed tools, without the legacy of the system codes or core solvers, or with more degrees of freedom when it comes to geometry and resolutions.

## 1.3 High-fidelity core simulations

Following the rapid development of the computational capacity, an increasing complexity in the fuel assembly designs and the continuous strive to perform more accurate and precise reactor core simulations, the last ten years have seen a large number of initiatives toward higher fidelity multiphysics simulations of the reactor core. From a larger perspective, the specific goals and aims are as many as the methodologies proposed and many of the initiatives are still in early development stages. Nevertheless, the trend is clear and true HPC applications are growing in interest in the nuclear community. In this section, some of the motivations of such novel schemes are discussed together with an overview of the HPC and high-fidelity efforts from the open literature.

### 1.3.1 Motivations for novel approaches

As evident from the discussion on the classical computational schemes, the separate neutronics and thermal-hydraulic reactor simulations are severely simplified in their representation of the multiphysics. As a motivation for new algorithms for nuclear reactor core simulations, a scattered list is given below. Some of the

points of this list are later referenced and rephrased in the actual objectives of the current thesis (Section 1.4):

- *Unresolved multiphysics* – The lattice codes apply simplified temperature and density profiles in the cross-section generation, implicitly introducing an error in terms of the actual thermophysical state in the fuel and moderator.
- *Small margins require higher resolution* – As plants are power uprated the margins to, for example, critical heat flux (CHF) decrease and arguably the local conditions are of increased interest [16].
- *Void heterogeneities* – Heterogeneities in the void fraction distribution in the subchannels of a BWR are potentially influential on the neutron moderation and thus a multiphysics understanding of such heterogeneities is of interest [17, 18].
- *Fuel behavior* – To simulate e.g. the local influence and deposition of CRUD on the fuel pins, novel fine-mesh and multiphysics schemes are required [19, 20]. Notably, simulation of CRUD deposition is one of the targeted problems in The Consortium for Advanced Simulation of LWRs (CASL) [21].
- *Spacer design* – As the spacers in the fuel assemblies increase in complexity, there is a greater need to understand the influence of the induced turbulence. In addition, CFD predictions of spacers are a cheap way to select the best candidates to a significantly smaller cost than using test rigs. [16, 22–24]

Whereas the list is by no means exhaustive, such and similar issues are interesting and constitute relevant drivers for the development of novel multiphysics and high-resolution strategies.

### 1.3.2 Overview of multiscale and multiphysics approaches

The drive towards stronger coupling and higher resolutions has taken a lot of different shapes, partly due to the fact that the complexities of the physics are mirrored in the complexities of the computations, partly because of unclear standards for reactor core multiphysics and partly because of the very different needs for the different issues mentioned in the previous section.

In many of the attempts on achieving the higher resolution multiphysics of the core, the schemes were based on externally coupled tools, e.g. applying a combination of a Monte Carlo solver for the neutronics and a commercial CFD solver for the thermal-hydraulics [25–27], or a deterministic neutronic solver [28–30], again, coupled to a CFD solver. The sophistication of such couplings varies from efficient data exchanges using external scripts to built-in coupling schemes. Large scale projects such as VERA [31] and MOOSE [19] likewise apply an existing code coupling approach but in massive HPC environments and with a large focus on

efficient parallelization and coupling. Although many of the efforts referenced are of interest from a physics point of view, many should be considered as proofs of principle, paving the way towards practical use of such computations.

In contrast to the external code couplings, there are notable examples of multiphysics solvers with a more integrated focus. An interesting example is again the MOOSE project, which can be employed as a general finite element framework with non-linear solvers and massive parallelization [32]. Many smaller scale projects with a tighter coupling were focused on commercial multiphysics solvers [33, 34], but there are also examples which are based on open source software [35], as is the case in the current thesis. Many of the multiscale and multiphysics initiatives in the open literature are focused on fuel performance in terms of local phenomena, where for example the deposition of CRUD is of importance [19, 20] or for fuel pin mechanistic behavior [36, 37].

Whereas the resolution of the CFD simulations has generally been low in the multiphysics coupled simulations discussed above (see e.g. [28, 30]), there are many pure CFD applications where a high-resolution approach is of particular interest. An example is seen in the simulation of grid-to-rod fretting for which time and space resolved turbulent fluctuations are of interest [22, 38]. Another class is the simulation of fuel pin spacers, where again an accurate prediction of the induced turbulence is of interest to predict the pressure drop in the core as well as the local heat transfer from the fuel pins to the coolant [16, 24, 39].

As mentioned above, multiphase flow is yet another topic for future high-fidelity simulations relevant for nuclear core predictions. Whereas the system codes rely on approximations and empirical relations, there are efforts made to perform 3D simulations on the scale of a fuel assembly (see e.g. Lo and Osman [40]). Although much less developed, there are examples of coupled multiphase CFD and neutronics [41], performed on coarse meshes and for steady-state purposes. However, the severe complexity of the multiple flow regimes makes the multiphase problem a theoretically more complicated CFD problem compared to the single phase counterpart, and much work remains in covering all flow regimes in a consistent manner [42]. An extended introduction connected to multiphase simulations for BWRs is postponed to Chapter 6.

## 1.4 Objectives of the research work

The objectives of the research are divided in two parts. First, the objectives for the fine-mesh multiphysics simulations are described together with the applied assumptions and resulting limitations. Second, the objectives for the investigations and method development for the multiphase flow research are outlined.

### 1.4.1 Fine-mesh multiphysics simulation objectives

The first part of the thesis is concerned with multiphysics simulations of the nuclear reactor core. The primary aim is to develop a computational tool which resolves the multiphysics dependencies already at the finest simulated scales. The physics covered is the same as in the classical coupled schemes (Section 1.2.3), i.e. the flow of the fluid coolant, the conjugate heat transfer (CHT) between the solid fuel pins and coolant and the neutron distribution in the core. However, the simulated scales are much finer, focused at a sub-fuel pin level and with a resolved water temperature and flow profile between each of the separated fuel pins. The described scales are throughout the thesis described as *fine-mesh*.

A part of the objective is to demonstrate the feasibility of such a computational code, including the computational cost and applicability in terms of HPC resources. However, the focus is also to resolve aspects of the multiphysics which are of potential importance for a safe operation of the reactor and, perhaps even more, for design of fuel assemblies.

In specific terms the objectives of the first part are to:

- develop a deterministic computational methodology with a fine-mesh approach to the neutronics for both steady-state and transient simulations, with a cross-section model relevant on a sub-pin scale,
- develop a CFD methodology, including heat transfer and the fluid flow, with a 3D representation of the flow between the fuel pins and treatment of turbulence,
- implement the methodologies in a single, multiphysics, computational tool deployable at computational clusters, and
- apply the solver to both steady-state and transient cases for parts of fuel assemblies.

As such, the objectives are related in a generic sense to many of the points mentioned in Section 1.3.1, and perhaps primarily to the need to resolve the multiphysics and to increase the resolution of the numerical predictions in the core. In relation to the previously referenced literature on coupled neutronics/CFD projects (Section 1.3.2), the current objectives are different in that a single computational tool should be developed, directly treating all covered aspects.

### Assumptions and limitations

Following the defined objectives, there are a number of implicit assumptions in the models, whereof the most important include:

- *No mechanistic modelling of fuel behavior* – As described in Section 1.3.2, many of current high-fidelity simulation schemes are primarily targeted toward

fuel behavior and mechanical properties. The fuel pins are here considered static, rigid bodies with only temperature dependent thermophysical properties.

- *No models for the surrounding environment* – Due to the computational cost (and the limited computational resources at hand) only parts of a fuel assembly are considered in the performed simulations. In principle a coarse model for the surrounding would be beneficial, serving a set of boundary conditions. However, for the presented cases all computations assume an infinite lattice of the simulated environment, which is realized through periodic or symmetry boundary conditions in horizontal direction. For the inlet and outlet conditions the boundary conditions are determined without any actual models of the nozzles or e.g. the turbulent spectra of the fluid flow entering the bottom of the assembly.

Whereas the previous assumptions are limitations of the implemented code, there are some further limitations imposed due to the computational cost of the simulations:

- *System size limited to parts of a fuel assembly* – As a result of the high resolution of the simulations, the computational grids contain a large number of degrees of freedom and thus an extensive computational burden. Due to the available computational resources, the system sizes must thus be limited. However, as part of the objective to implement the tools for HPC environments, proper parallelization schemes are still applied and with larger computational resources the code should be applicable to larger cases.
- *Number of neutronic energy groups limited* – The neutronic calculations are performed on a low number of energy groups (ranging from 2 to 16). Again the reason is the computational cost, and all algorithms are implemented for an arbitrary number of energy groups (an exception to this is the results related to Paper VIII, where the derivation is performed for 2 groups (Section 5.3). Similarly, the simulations performed with the discrete ordinates method are limited in the number of directions.

Practically there are additional limitations in terms of what could be considered within the above scope of the objectives. In particular, the geometrical details of the fuel assemblies are limited to the fuel pins, the gap, the cladding and the coolant, e.g. neglecting the influence of the spacers.

## 1.4.2 Multiphase flow objectives

The second part of the thesis concerns multiphase flow in the reactor core and in particular bubbly flows under subcooled and adiabatic conditions. As is further discussed in Section 6.1, the multiphase flow is challenging for multiple reasons

and the maturity of multiphase CFD is significantly lower than for single-phase. For the mentioned reasons, the research conducted for multiphase flow within this thesis has a more generic character than the fine-mesh multiphysics simulations. Nevertheless, the goal of the conducted studies is to increase the understanding of bubbly flows with the two-fluid model, which could be used for low void fraction simulations in the fine-mesh multiphysics solver.

Due to the extreme computational cost connected with the interface resolving methods, such methods are not of practical interest for simulations of systems of the size of a fuel assembly. Instead, the focus is on the two-fluid model, which gives only an average representation of the phases, with no explicit tracking of the interface between the gas and the liquid in the two-phase flow. The model is further introduced and discussed in Section 6.4, but for the sake of formulating the objectives for the multiphase research two of the issues with the two-fluid model is here outlined.

First, as a result of the lack of representation of the interface between the faces, information such as bubble sizes and shapes are unknown in the two-fluid formulation. A potential remedy for this is to introduce a population balance equation (PBE) to track one or more properties of the bubbles to a significantly lower cost as compared to explicitly computing the bubble interfaces. Such an approach is of interest not the least for diabatic simulations where the bubble distribution will change not only due to bubble breakage or aggregation but also due to condensation and evaporation. In practical terms, a two-fluid approach complemented by a PBE is a good candidate for simulation of the subcooled and bubbly flow regimes in a BWR, and as such the framework is worth investigating, also from the fine-mesh multiphysics point of view.

Secondly, from a more general perspective, the two-fluid formulation has previously been shown to be prone to numerical issues, not the least due to an apparent lack of hyperbolicity for some types of flow. Although a wide range of remedies have been proposed, the underlying potential stability issues are still of major interest for the application of the two-fluid model. In particular, the dynamic behavior of the model is key to the predicted mass and heat transfer within the reactor core, and emergence of heterogeneities in the flow can only be trusted if the underlying equations are understood to be sound. Again, investigations on the dynamics of the two-fluid formulation is of interest for the larger perspective of fine-mesh simulations as the dynamics and appearance of void heterogeneities are of potential importance in the coupling to the neutron distribution.

In specific terms, the objectives related to modeling of multiphase flow within this thesis are to:

- investigate the stability of the two-fluid model in terms of phase heterogeneities for bubbly gas-liquid flows, and
- develop novel methodologies for simulation of vapor bubbles in a subcooled liquid using a the two-fluid model complemented by a PBE.

As the case of multi-phase flow is particularly challenging from a CFD perspective and again limitations to the scope are necessary. The main simplifications applied in the developed models and performed simulations include:

- *No simulation of wall phenomena* – The bubbly flows arises due to evaporation at the wall. The bubble growth and departure are governed by micro structures at the surface as well as the flow at the wall. Such aspects are not covered in the thesis. Instead the transport of the bubbles is the primary target.
- *Empirical correlations for condensation* – In the simulation of subcooled bubbly flows, the condensation of the bubbles is, again, ruled by local conditions at the surface of the bubble. In this work no interface tracking simulations of the condensation are performed. Instead, empirical relations are used for the condensation rate.
- *No specific treatment of high void fractions* – The objectives are primarily related to low void fraction regions. This is not to say that high void fractions with other regimes than bubbly flow are not of interest. Rather, the bubbly flows are perhaps the most well researched and still there are significant opportunities for further research. From a larger perspective it is also natural to first focus on the low void fraction regimes.

### 1.4.3 Outline of the thesis

The thesis is structured in seven chapters, whereof this introductory chapter is the first. Chapter 2 gives an introduction to the computational methodologies later applied in the thesis, including detailed descriptions of mesh generation, cross-section generation and parallelization algorithms. In Chapter 3 the models for coupled LWR single phase problems are given for both neutronics and thermal-hydraulics. Next, Chapter 4 presents the implementation and an application to a steady-state multiphysics problem. In Chapter 5, the methodology is extended to transient conditions and again applied to a small lattice of fuel pins. Chapter 6 describes some of the complexities of the two-phase flow in BWRs, and presents a proposed algorithm for subcooled boiling and, additionally, simulations regarding the dynamics of two-phase CFD solvers. Finally, Chapter 7 gives a conclusion and recommendations for future work in the areas of multiphase flow for reactor core applications as well as fine-mesh multiphysics simulations.





# Computational methods

To perform multiphysics simulations of a nuclear reactor is, inevitably, strongly connected to development of computational methods, coding, and not least HPC. To approach both the complexity and the sheer size of the system, we need efficient algorithms and numerical methods run on modern hardware and implemented in the right languages. I therefore find it reasonable and enlightening to introduce the computational techniques, which are key to this thesis. As a matter of fact, a large effort has been invested in choosing and developing sufficiently performant algorithms; a task which has been equally challenging and joyful.

Due to the many fields of physics and thus many numerical solvers required, it would be difficult, not to say impossible, to within the same PhD project develop all the necessary computer code from scratch. Instead, I partly use some existing tools and codes, in some cases extended for the purpose of the project, which substantially increase progress and reduce the development time. At the same time, it is of large value to have full transparency of all codes and algorithms. With the current trend of open-source initiatives this is viable. An accessible code base gives the possibility to perform rapid development and the opportunity to modify and extend the software. This has been a cornerstone in the work for this thesis.

This chapter introduces the computational tools applied throughout the remaining chapters of the thesis. A brief introduction to HPC is given in Section 2.1, including a small historical perspective on HPC and the development of computers and efficient code in parallel algorithms. In Section 2.2, an outline of the key elements of the finite volume method (FVM) is presented together with the library which lies as the foundation for the developed multiphysics solver as well as the two-phase solvers later discussed. Finally, Section 2.3 introduces the specific framework developed within this thesis and its use of existing software.

## 2.1 High Performance Computing

High performance computing involves simulations or other computational tasks that are employed on multiple processors or multiple computers. In the area of computational physics and numerical simulations in general, HPC is necessary to solve large problems which would lead to prohibitive long simulation times on a laptop or desktop computer. Now, as we shall see, the notion of a large computational problem has changed and continues to change with the ever increasing capacity of the supercomputers. The computations performed in this thesis would have been almost impossible already 20-25 years ago, even considering, at that time the largest supercomputers in the world.

To further widen the concept of HPC, aspects such as computational efficiency and utilization of the hardware must also be considered. In particular, even though a large computational cluster is exploited, the single CPU utilization is still vital. Furthermore, the single CPU optimization is relevant irrespective of the cluster size. As discussed below (see Figure 2.1), the computational resources for this project have been limited to around maximum 80 and in average 20 CPUs on one of the Swedish computational resources.

### 2.1.1 A brief historical perspective

The history of HPC and supercomputers goes back to the 1960s, with computers such as LARC [43], featuring two CPUs. The early 1970s saw the emergence of RISC computers, where the CDC6600 was the first to use the idea of a reduced instruction set to simplify the CPU [44]. From the later part of the 1970s, HPC was dominated by vector computers which utilized a single operation on a vector of data (so called SIMD architecture) [45]. During the following two decades (ca. 1975-1995) the supercomputer CPUs were considerably more complex than personal computers. In the latter half of the 1990s this changed, with the emergence of clusters built up of thousands of simpler, commodity, CPUs. During the 2000s the trend with massively parallel supercomputers continued, with machines such as Blue Gene [46].

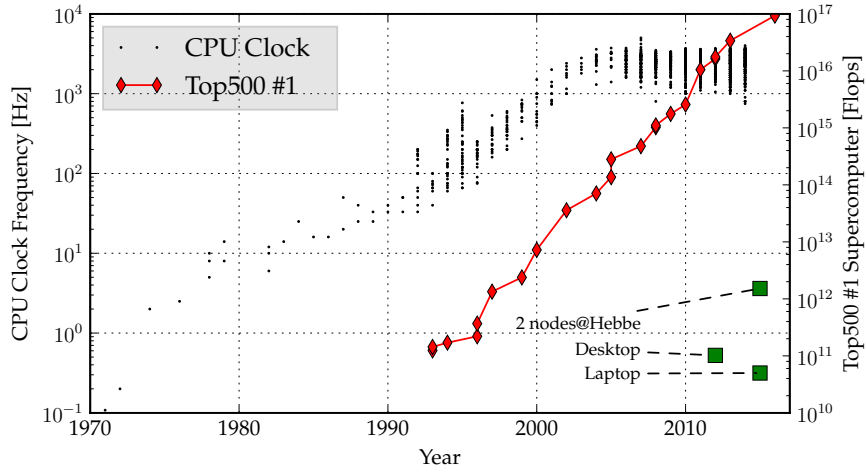
Although history has seen a large number of architectures of hardware the trend is clear; the computational capability is growing exponentially (see Figure 2.1). With the trend of ever increasing computational capacity it is tempting to rely on accelerated hardware, putting less effort in software development. However, in reality, the situation is rather the reversed one. Currently, the trend of the largest supercomputers is to use more heterogeneous architectures with accelerators or graphic cards connected to each standard computer. To use the full capacity of such resources major efforts on software development might be required to utilize the resources.

From a nuclear engineering perspective, it is interesting to note that, whereas

the nuclear industry for a long time has taken a lead in the development of HPC resources, Turinsky [47] points out that during the mid 1980s the industry turned from the use of large computers to standard desktop machines. Although the industry to a large extent relies on codes running on desktop computers, many of the above described efforts (Section 1.3) are again heavily focused on taking the lead in use of large scale computations.

#### Hardware development and resources

The trend for the growth of the supercomputers is shown in Figure 1, with a linear increase in the floating point operations per second (FLOPS) in the log diagram. In relation to the single CPU performance, the increase in clock frequency of the CPUs is also important. However, as indicated by the trend in the figure, the growth in clock frequency has stalled and we cannot expect the next generation of CPUs to necessarily keep accelerating our applications. To quote Herb Sutter, the convener of the ISO C++ committee, "The free lunch is over" [48].



**Figure 2.1:** Development of the CPU clock frequency over time [49] (data extracted from [50]) with a comparison of the #1 computing cluster according to Top500 [51]. The green squares indicate desktop and laptop computers of the thesis author, with Flops estimated by Intel Math Kernel Library Benchmarks [52].

### 2.1.2 Code efficiency and optimization

To write a fast computational code we need to consider both hardware aspects, such as e.g. cache sizes and memory bandwidth, as well as software aspects such as algorithms and code languages. Although a PhD project in multiphysics of nuclear reactors does not specifically target such areas, it is arguably of high value with a general knowledge of such code aspects for the development of a HPC framework.

In order to illustrate some of the many challenges in code efficiency and optimization, a few of the key aspects for the multiphysics simulations performed are briefly mentioned:

- *Memory bandwidth* – To solve the discretized neutronic and thermal-hydraulic equations, a sparse matrix solver is applied (Section 2.2.2). The standard iterative sparse matrix solvers typically rely on matrix-vector multiplication, which in turn are limited by the bandwidth of the memory [53]. Whilst the modern CPU architecture relies on multiple levels of caches to speed up the data fetching, our large sparse systems will not be close to fit the cache.
- *Memory locality* – Not only sparse matrix calculations are restricted by slow memory access in relation to the CPU throughput. For data processed together it is always beneficial to keep the data close in memory. An important example is the ordering of the computational mesh cells which could help to minimize the number of fetches from memory (and the larger cache levels).
- *Disk access* – The disk (spinning disk or SSD) is the slowest data storage on the computer and we generally want to avoid excessive write or read operations. As an example, we can avoid disk access by condensing the results as much as possible already in the simulation stage, minimizing the storage operations.
- *Vectorized operations* – A modern CPU support vectorized operations where multiple variables are processed within the same instruction (see e.g. [54]). On a commodity CPU this means that the throughput of arithmetic operations for double precision (64 bit) numbers increases with a factor of two to eight (see e.g. [55]).

For any of the above aspects we are helped both by the CPU routines (e.g. for handling data fetching) and the compiler (e.g. for translating code in to vectorized machine instructions). Nevertheless, it must be in the interest of the developer to understand the basics of the hardware and software interaction. Any small knowledge helps to increase the understanding of the computer and what we can expect in terms of CPU efficiency and performance.

Directly or indirectly, the programming language is an additional key aspect for good utilization of the hardware. HPC applications have historically been implemented in compiled languages, where no additional effort is spent on run-time code interpretation. Whereas FORTRAN is a classical choice for HPC for nuclear applications in particular, a large proportion of modern software is developed using other languages, not seldom C++. The latter has the advantage of a large user base (outside the nuclear community and outside HPC) with good compiler and library support.

### 2.1.3 Parallelization

A fast single CPU code is a good starting point for a fast parallelized code. Modern architectures support parallelization on many different levels, which might

require significantly different efforts from the developer. The three most commonly discussed parallelization regimes include:

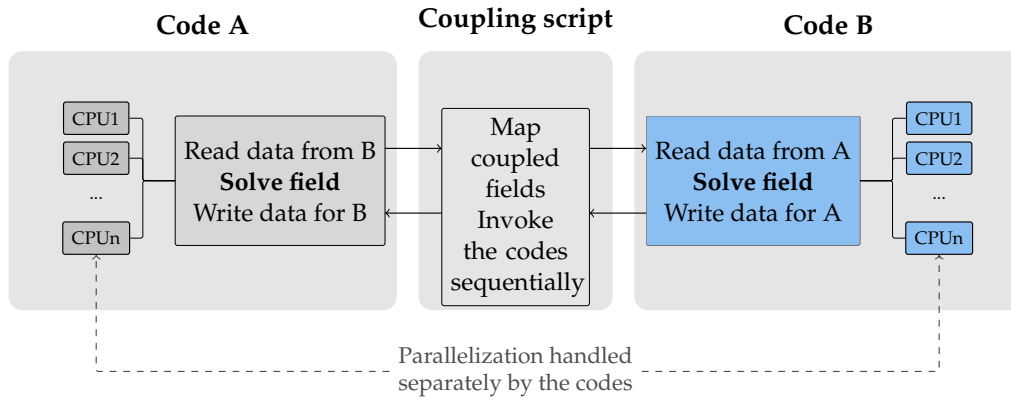
- *Shared memory* – The easiest way to parallelize the code is often to use all available threads on the same computer, which allows the memory to be shared between the different processes. Shared memory parallelization is limited to the number of CPUs on the motherboard (i.e. 2-6 on a commodity desktop computer). A major advantage is that the interprocessor communication is avoided, i.e. the parallelization incurs no substantial overhead.
- *Message passing* – To combine multiple separate computers (aka nodes) on e.g. a computer cluster, message passing is utilized. In contrast to the previous scheme, no shared memory exists and the application must handle exchange of all data common between the different computers. While this is de facto standard seen in e.g. commercial CFD solvers (as well as the open source alternatives) it has drawbacks in terms of interprocessor communication overhead and complexity of data exchange scheduling.
- *Accelerators/Graphic cards* – A recent trend on some of the largest supercomputers has been to use separate hardware for computational acceleration. For example, graphic card computations based on, e.g., CUDA [56] is a current trend in which massive computational performance can be achieved on a single gaming graphics card. The major drawback here is the induced complexity of another hardware architecture.

In addition, for completely independent simulations, we can of course parallelize by running multiple separate processes and combine the results. The latter is e.g. used in some Monte Carlo codes, and was exploited in the cross-section generation (Section 2.3.3), where the system can be duplicated since the neutron histories are assumed independent. A further detailed introduction to different types and levels and some key aspects for nuclear engineering parallelization is given by Calvin and Nowak [57].

In the same manner as the above discussion on efficiency and optimization, the code developer will benefit from general knowledge about the parallelization. However, in many cases an underlying framework for parallelization is desirable, e.g. a parallelized matrix solver library. In the current project the parallelization is achieved using message passing as defined in the MPI standard [58]. The system is decomposed in multiple domains, each solved in a separate process. The message passing handles the communication by, simply put, exchanging the boundary values of the domains.

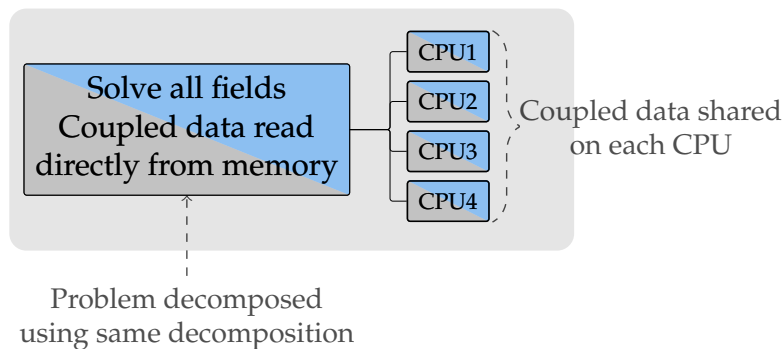
In more specific terms of reactor core simulations, many of the CFD/neutronics multiphysics examples from the literature are formulated as a multiple code scheme. Such an approach is exemplified in Figure 2.2. Advantageously, each of the codes can use the existing parallelization capabilities, which is efficient from a development point of view. However, as apparent from the figure, the communication is handled in a gather-scatter manner, where the internal parallelization, e.g. by

domain decomposition, must gather data before the exchange to the other code occurs. After retrieving the data from the other code a scatter must again take place. This clearly limits the efficiency of the coupling and induces further complications [31].



**Figure 2.2:** Example of data transfer in the multiple codes approach [Paper IV].

In contrast, in a single code approach, as applied in the thesis, all parts of the multiphysics problem can be decomposed in the same spatial domains as exemplified in Figure 2.3. In such a scheme, the data transfer between e.g. the neutronics and the thermal-hydraulics can be performed directly on each separate CPU or thread. As the same, monolithic, application runs both the CFD and the neutronics there are no costs associated with waiting for one of the fields to be finished or similar. Instead all fields are directly solved in the same code. This is particularly important for transient applications where a large number of data exchanges are required.



**Figure 2.3:** Example of single code coupling scheme avoiding external transfer [Paper IV].

## 2.2 The finite volume method

In this thesis all partial differential equations (PDEs) are solved using the finite volume method (FVM). FVM is the classical choice for CFD applications, including many commercial examples (e.g. Star-CCM+ [59] and ANSYS Fluent [60]), as well as long term open source projects (e.g. Gerris [61], OpenFOAM® [62] and MFIX [63]). Although the finite difference method (FDM) was the historical choice, in particular for the very early CFD research [64], it is much less popular today. FDM suffers from restrictions on the grids and the conservation properties are less favorable than in FVM [65]. Furthermore, the finite element method (FEM) is applied in some CFD codes (e.g. COMSOL [66]), but still only in a minority of the commercially available tools. FEM has undoubtedly many advantages, not the least for higher order discretization, but still to some extent considered less mature than FVM for fluid systems [67].

In terms of 3D implementations of neutronics (i.e. core solvers), the classical choices include the FDM or nodal methods [68]. However, in terms of differential operators in the neutron diffusion and transport equation, the FVM is again a viable option. To fulfill the objective of a single multiphysics code, the neutronic equations are here solved with the FVM, as done for the CFD. In particular, FVM is well suited for unstructured meshes required in the fine-mesh approach in this thesis.

### 2.2.1 Equation discretization

The finite volume method relies on a discretization of the computational domain into control volumes (CVs). The process of discretizing the domain is considered separately below (Section 2.3.1) and at this point only the discretization of the conservation equations on an existing grid of CVs is considered. Even though a full description of the discretization mathematics is out of scope for the current thesis, a brief and very basic overview is given. A more complete and general picture can be found in numerous CFD books and publications (for good examples see e.g. [65, 69, 70]).

For simplicity we consider a standard example from CFD, namely the transport equation of the generic (scalar) quantity  $\theta$ :

$$\frac{\partial \theta \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U} \theta) - \nabla \cdot (D_\theta \nabla \theta) = S_\theta(\theta), \quad (2.1)$$

which is transported in a fluid by a convective velocity  $\mathbf{U}$ , with a fluid density  $\rho$ , a diffusivity  $D_\theta$  and some source term  $S_\theta$ . For the sake of brevity, notation on the space and time dependence has been removed from all quantities in eq. (2.1). Upon integration over the CV and after applying the Gauss theorem we get

$$\int_V \frac{\partial \theta \rho}{\partial t} dV + \sum_{\forall f} \mathbf{S}_f \cdot (\rho \mathbf{U} \theta)_f - \sum_{\forall f} \mathbf{S}_f \cdot (D_{\theta,f} \nabla \theta_f) = \int_V S_\theta dV, \quad (2.2)$$

where the divergence operator convective term and the diffusion term are transformed to discrete sums over the faces of the control volume by application of the Gauss theorem.

Concerning the space dependence, FVM assumes a linear variation of a field both with respect to time and space and, for the collocated approach, cell centered values of all fields [70]. Accordingly, the volume integrals in eq. (2.2) are converted to multiplications of the integrand and the volume of the current cell. For the right hand side source term a linearization in the field can be performed, increasing the implicitness of the method, such that

$$\int_V S_\theta(\theta) dV = V_p \theta_p S_p + V_p S_u, \quad (2.3)$$

where  $p$  indicates the cell currently under consideration,  $S_p$  and  $S_u$  correspond to the coefficients for the implicit and explicit parts of the linearization, respectively. This is of particular interest for the implementation of the neutronics, where linearizations of both the fission and scattering source terms are needed.

For the convective term, face values of the velocity, density and the generic quantity are required. The interpolation to faces is of major importance for the accuracy as well as the robustness of the method. Whereas low order methods, such as the upwind differencing guarantee boundedness, higher order schemes such as central differences come at the expense of issues with stability [69].

Similar to the convection, the diffusion term requires a face interpolation of the density, the diffusivity and the gradient of the field ( $\nabla\theta$ ). For orthogonal meshes the gradient can be directly estimated as the difference between the cell center values in the cells on each side of the face. For non-orthogonal meshes on the other hand, additional explicit terms are needed (for a detailed overview see [70]).

To handle the time dependence, eq. (2.2) is integrated in time. The time derivative is then approximated as

$$\int_t^{t+\Delta t} \int_V \frac{\partial \theta \rho}{\partial t} dV dt = \rho_p \frac{\theta_p^n - \theta_p^o}{\Delta t} V_p, \quad (2.4)$$

where the indices  $n$  and  $o$  refer to the new and the old time step, respectively. What concerns the rest of the terms of eq. (2.2), for all variables a time step ( $n$  or  $o$ ) must be chosen. If the old time step is inserted an explicit scheme is achieved, whereas using the new time step (currently solved for) results in an implicit scheme. The previous is first order accurate and limited by the Courant number whereas the latter is in theory first order accurate but unconditionally stable. In practice however, the first order accuracy of both schemes are undesirable and therefore the unconditional stability is of secondary importance. Other, higher order time schemes are therefore more popular, and in the current thesis the default choice is the Crank-Nicholson scheme (second order in time).



For later discussion on mesh generation (see Section 2.3.1) the following conditions imposed by the applied FVM should be noted:

- *Only first neighbors* – In the discretization of the convective term and the diffusion term relations to neighboring cells are computed. In the current work only first neighbors are considered, resulting in a so called compact computational molecule. The latter is beneficial for unstructured meshes, where second neighbors are not trivially defined.
- *No hanging nodes* – The faces of the CVs are assumed to be one-to-one in the sense that no single face is connected to more than one face of another cell.
- *No curved faces* – All faces of the discretized mesh are assumed to be flat. A curved boundary (such as the outer radius of the fuel pin) is thus represented by a set of flat faces. In contrast to FEM, where the order of accuracy can be increased by an increase of degree of base polynomials and with a non-linear representation of the boundary (see e.g. [71]), FVM instead relies on a refinement of the mesh.

Although seemingly restrictive, the above assumptions are important to achieve a performant method with fast discretization and a, generally, well structured sparse matrix system with few off-diagonal elements.

## 2.2.2 Sparse matrix solvers

The discretization in time and space results in a set of algebraic equations. It should be noted that all non-linearities are linearized (as discussed for the source term), and thus result in a sparse linear matrix system.

In terms of computations, the discretization (and matrix assembly) routines might take significant computational time. However, the major effort of the CFD solver is spent on solving the linear system itself. Whereas direct methods are useful for very small matrices, such methods are out of question for the large and sparse matrices found in CFD, both due to excessive memory usage and too high a cost in terms of floating point operations. Instead, iterative methods are used such as e.g. the Gauss-Seidel method for which a huge number of acceleration techniques have been developed over the years [72]. Again more efficient methods are found in so-called projection methods, which include Krylov subspace solvers such as Conjugate Gradient (CG) for symmetric matrices and Bi-Conjugate Gradient Stabilized (BiCGStab) which are routinely used in many CFD solvers. For certain classes of problems, relaxation techniques, such as the Algebraic Multigrid method (AMG), are a good option.

Although the mathematical details of the sparse matrix solvers are out of scope for the thesis (for good introductory texts see e.g. [73, 74]), a general knowledge is of importance both for selecting the correct class of methods based on the matrix properties (hyperbolic, parabolic or elliptic) as well as for tuning the

methods to achieve good performance. The latter is often a (time consuming) process of trial and error. Nevertheless, due to the long simulation times for highly-resolved simulations presented in this thesis, well chosen parameters can easily save days of simulation time.

In the presented example (eq. (2.1)), the solution variable  $\theta$  was assumed to be a scalar quantity. For vectorial (or tensorial) solution variables a set of such conservation equations need to be solved for each computational cell. Additionally, with interdependent conservation equations (such as pressure and velocity, or different neutron energy groups) the couplings must be resolved. In principle all equations could be discretized together, and after applying the required linearizations, a coupled system of equations is obtained. In CFD the early day computers were limited by computer memory and as a result, the matrices were kept at a minimum size, resulting in the development of segregated algorithms for the pressure and velocity dependence. The coupled aspects are further discussed in Section 3.3.

### 2.2.3 OpenFOAM®

All solvers developed in this thesis are implemented as extensions of the finite volume framework OpenFOAM®[75]<sup>1</sup>, which is a C++ open-source tool for development of CFD solvers. The code consists of a library of general routines focused on formulation, discretization and solution of tensorial equations with FVM. Furthermore, a set of existing solvers are provided, including classical approaches such as segregated pressure and velocity solvers for single phase flow, multi-fluid solvers, conjugate heat transfer and more. For an overview, see the documentation [77, 78] or third party literature on the code [79, 80].

Considering the objectives of this thesis, some of the key aspects and benefits of the library are:

- *Unstructured meshes* – OpenFOAM® handles unstructured meshes and implements a computational molecule consisting of the nearest neighbors. The unstructured meshes captures the geometry of the fuel pins and potentially more complicated core geometries (e.g. the spacers). The library provides corresponding FVM discretization and differential operators for the fluid flow and heat transfer applications.
- *High level equation format* – The code provides a high-level equation interface in which the system equations can be directly written in a user-friendly manner which effectively hides the underlying equation discretization and treatment of boundary conditions, etc.

---

<sup>1</sup>OpenFOAM® has since its first release in 2004 been forked multiple times, and in this thesis the multiphysics code and the subcooled population balance transport code are implemented in the community version foam-extend [76] whereas the two-fluid stability computations are performed in OpenFOAM foundation version [62].

- *Existing CFD algorithms* – Importantly, the library provides the standard segregated fluid solver approaches, including SIMPLE [81] and PISO [82], with a Rhie-Chow for collocated grid [83]. The previous are the foundations for the fluid solver for the coolant of the core.
- *Turbulence models* – Libraries of both Reynolds Averaged models (RANS) and Large Eddy Simulations (LES) are included, which enable fast integration of turbulence in new solvers.
- *Parallelization routines* – The library supports MPI parallelization and all necessary data operators (such as discretization routines, algebra operators) are written with the parallelization in mind. The code uses a so-called zero halo layer domain-decomposition, i.e. partitioning of space in non-overlapping domains.

Except for the above mentioned points, the freely available source code (released under GNU GPL 3) is another important aspect as it can be extended not only on the top-level but also for key, lower level, elements of the library.

For this thesis, many parts of the library are exploited without any modifications. However, for the multiphysics and two-fluid parts of the project, significant code development was needed. Some of the newly developed parts<sup>2</sup> are (in order of size of code):

- *Neutronics solvers* – A library for neutronic simulations (diffusion and transport) was required. This includes handling of the cross-sections and treatment of both diffusion and transport equations ( $S_N$ ).
- *LWR multiphysics solver* – The multiphysics application was developed to combine neutronic solvers and fluid solvers.
- *Population balance solvers* – For Paper VI, PBE solvers for DQMOM and MUSIG were developed, including the relevant boundary conditions for vapor generation.
- *Mesh intersection utility* – A library for intersecting overlapping meshes was developed. The tool handles the data transfer between the neutronic mesh and the separate meshes for the solid fuel pins and the coolant (see Section 2.3.1).

---

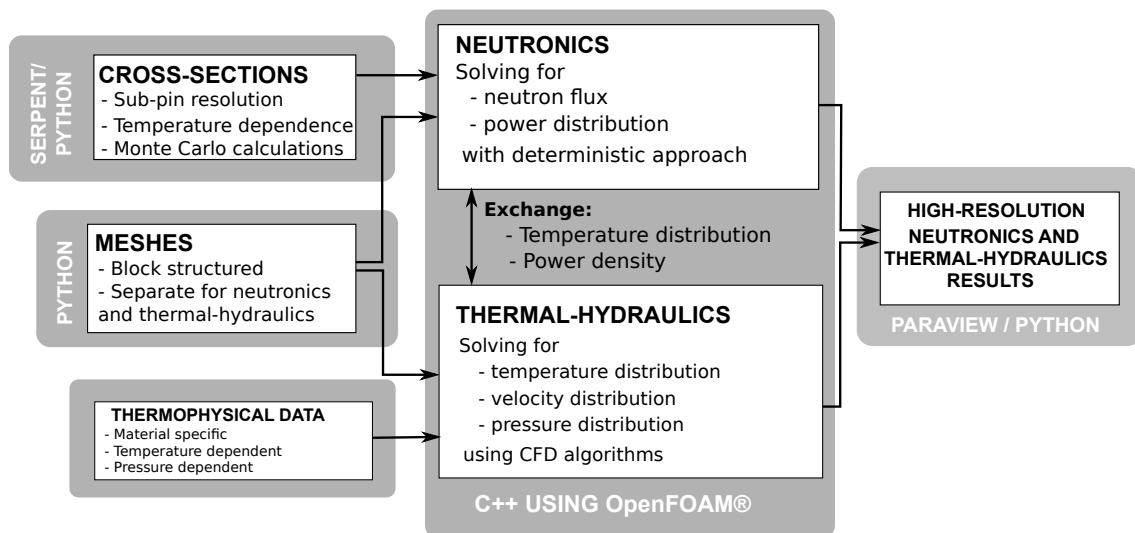
<sup>2</sup>The mentioned additions to the code are not part of OpenFOAM®, and the work here presented is not approved or endorsed by OpenCFD Limited, the producer of the OpenFOAM software and owner of the OpenFOAM and OpenCFD trade marks. For any ambiguity on the rights (and wrongs) of the trademark please refer to <http://www.openfoam.com/legal/trademark-guidelines.php>.

## 2.3 Computational scheme

An overall view of the computational workflow of the multiphysics solver is presented in Figure 2.4. The leftmost column corresponds to the pre-processing steps, namely mesh generation (Section 2.3.1), cross-section generation (Section 2.3.3) and formatting and acquiring of thermo-physical data for the materials. As for the material properties, data are extracted from openly available sources and interpolated in the multiphysics solver.

The center column corresponds to the coupled fine-mesh solver for the neutronics and the thermal-hydraulics. As indicated, the two modules exchange all coupled data, and the procedure for the data mapping is briefly covered in Section 2.3.2. The results of the computations are stored on the OpenFOAM internal format and post-processed by a combination of ParaView [84] and Python utilities.

Although invisible in all included publications and in the current description, the setup of a coupled case with all initial fields, meshes, thermophysical data, cross-section sets and decomposition schemes is typically time-consuming and repetitive. To simplify the process, Python utilities were developed for each of the applications presented in the appended papers, and the value of such utilities for work like this should not be underestimated.



**Figure 2.4:** Overview of the computational framework for the multiphysics simulations.

### 2.3.1 Grid generation

The generation of computational grids, is a key to the multiphysics simulations, not least due to the fact that the problem benefits from the use of multiple meshes, with one or more for each field.

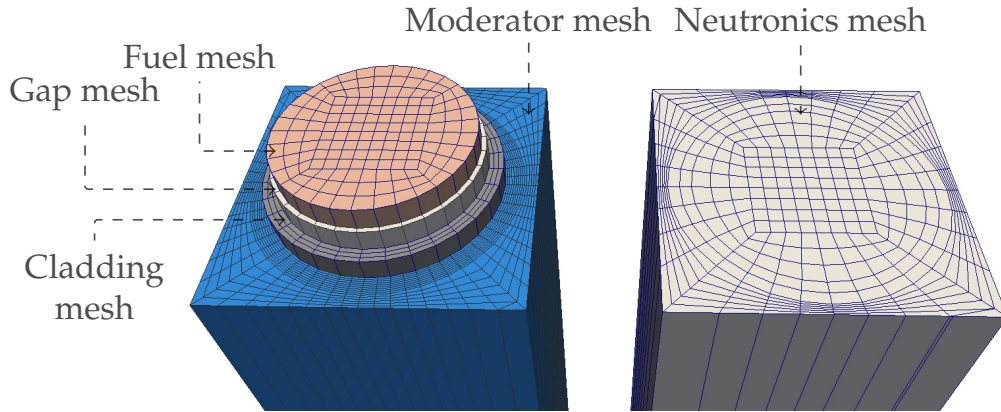
#### Mesh influence on the results

As regards the influence of the grid quality on the results, some of the key sources which contribute to mesh-induced errors are [70, 85]:

- *Insufficient mesh resolution* – If the mesh is too coarse the physics might not be correctly represented. In particular, large gradients will be poorly represented and the solution might not be only imprecise but also inaccurate.
- *Non-orthogonality error* – In the discretization of the diffusion operator, the gradient at the surface between two neighboring cell centers must be computed. In the case of non-parallel normal of the surface and the vector connecting the cell centers a correction of gradient at the surface is required. Such a correction is often computed explicitly [65], but might lead to instabilities.
- *Skewness error* – The term skewness is applied when the vector binding neighboring cells does not cross the common surface in the center. In particular, the order of interpolation from the nodes to the face centers decreases from second order to first order.
- *Non-uniformity* – The mesh uniformity affects the discretization, namely via the order of accuracy of the computation of face gradient.

In addition, it should be mentioned that in the case of a preferred direction of the flow (as the case with the dominating axial flow velocity component in the reactor) it is preferable to have the flow crossing the cells parallel to the normal of the crossed surface. In practical terms we thus desire prismatic or hexahedral elements, axially aligned with the flow.

As the gradients in the field govern what regions require a finer discretization, a non-uniform grid is generally beneficial. The resolution requirements would either stem from some a priori knowledge of the solution or on posteriori estimated errors used to automatically update the mesh accordingly (so called adaptive refinement). Whereas we can directly apply the former from a general knowledge of the reactor core, considering phenomena such as a thermal neutron group peak in the top and bottom reflectors, the latter is, although tractable, practically cumbersome to achieve. In particular, for a refinement by splitting cells to smaller cells (h-refinement) we want to work on a fully unstructured grid [86], which has other drawbacks for the multiphysics framework as discussed in the next section.



**Figure 2.5:** Example domain discretization for a single fuel pin with surrounding subchannel. Exploded for the thermal-hydraulics (left) and the neutronics (right) [Paper IV].

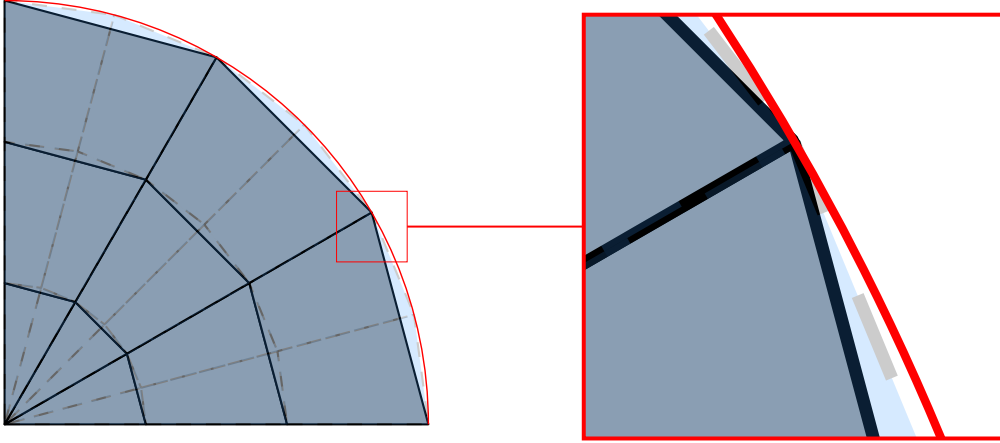
### Multiple meshes

To solve both neutronics and thermal-hydraulics in an efficient manner, multiple overlapping meshes are required. The neutronic grid must cover the entire domain, whereas the fuel, gap and cladding meshes discretize the separate regions only. In addition, the fluid problem is solved on a mesh covering the domain outside the solids. An example of the required regions and meshes is given Figure 2.5, where the separate domains are highlighted.

Depending on the algorithms that couple the different fields of physics, different conditions are imposed on the mesh characteristics. A good overview of some mesh generation issues specific to the reactor geometry is given by Hansen and Owen [87], where among others the following key aspects are notable:

- *Mass preservation* – A mesh with flat surfaces induces an error in the representation of the underlying body. Figure 2.6 gives an example where a fuel pellet is discretized with two alternative meshes; the full lines apply a discretization of 3 elements in the azimuthal direction whereas the dashed discretization uses 6 elements. In the magnification of the point at  $\phi = \pi/6$  it is seen that the actual cylinder (red) is not exactly captured by any of the two meshes. Although seemingly small, the impact on the criticality value of the reactor might be important.
- *Multiple mesh consistency* – Not many available meshing tools are built for multiple mesh support, especially not considering the need for overlapping meshes with preserved material regions. Preferably, a mesh generator should handle situations as the one presented in Figure 2.6 in a consistent manner for all the meshes.

In addition, two more aspects were found important during the method development for this thesis:



**Figure 2.6:** Quarter fuel pin discretization with a 9 elements (solid lines) or 18 elements (dashed lines).

- *Domain decomposition* – As all the calculations are performed on HPC resources with MPI, the domain must be decomposable into different regions. To achieve an efficient decomposition of the multiphysics problem, one approach is to decompose all fields of physics using the same spatial partitioning. This ensures that the data transfer between the modules (e.g. neutronics power level to the fuel heat transfer mesh) is done on the same CPU. To achieve this, a strictly controlled meshing process is required, where all the different meshes have cell faces coinciding at the surfaces used for spatial partitioning.
- *Mesh repeatability* – Since the geometry of the reactor consists of square or hexagonal lattices, at least in the case of the commercial designs, a consistent meshing throughout the fuel assembly is needed. That is, for the same pin cell geometry, the same mesh topology should be computed. This is achievable with block structured meshes where one can assure a perfect match at the interface of the pin cells. For an unstructured mesh this is not directly feasible (or at least difficult to achieve given the previous point of material region preservation).

The described key aspects are not very closely related to the archetype case of the single monolithic mesh covering the full domain. Although the mentioned issues are not particularly challenging from a theoretical point of view, the combination of requirements still makes for a practical problem requiring special procedures (in the sense of not directly applicable in the commercial CFD codes). As an example from the literature, Tautges and Jain [88] describe a hierarchal procedure for building meshes for hexagonal as well as square lattices with a sufficient resolution for CFD, taking advantage of the repeated structures. Such procedures are a good attempt, at least up until the point where the level of detail is further increased. For the case of explicit modeling of spacers, a fully unstructured mesh is generally unavoidable (see e.g. [40, 89]).

### Development of mesh capabilities

If we summarize the above discussion, the criteria include that the meshes should be:

- preferably structured with minimized non-orthogonality and surface normals aligned with the flow direction,
- conformal with the domain decomposition such that no inter-node data transfer is required for the multiphysics couplings,
- easy to control in terms of the resolution, in order to minimize the degrees of freedom,
- repeatable for all pin cells, and
- body-fitted to capture advanced geometries such as spacers.

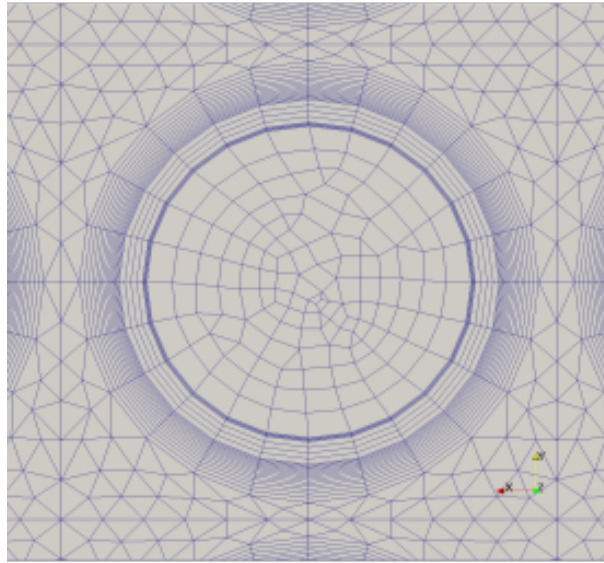
To fulfill as many criteria as possible, two different solutions were tested. In the first part of the project (essentially [Paper I]) the meshes were generated with the SALOME platform [90]. An example of the meshes produced is given in Figure 2.7. The SALOME platform includes both pre-processing and post-processing together with different capabilities for code coupling and was applied as the framework tool for the NURESIM project [91]. For the current example, only the meshing tools were tested.

The meshes from [Paper I] are of hybrid character, with structured regions for the boundary layers (i.e. fluid regions close to the walls) and unstructured regions in the bulk of the fluid and the fuel pin. Although the software could be controlled via a Python interface, the possibilities to full control the mesh was found limited (at least in 2012).

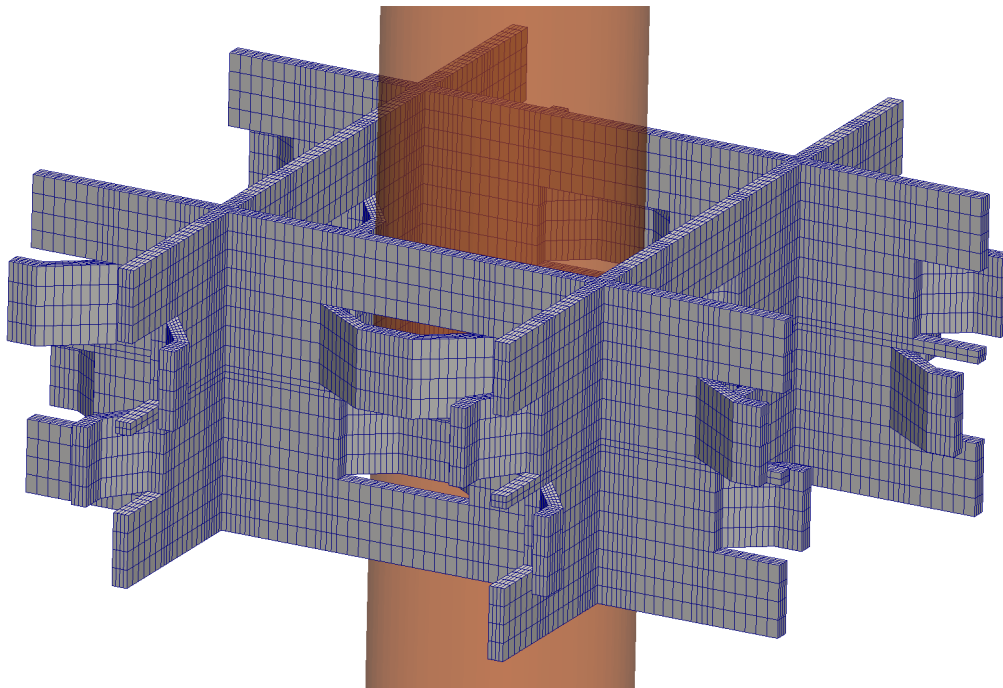
To increase control, especially to get full consistency between all mesh regions and in the whole fuel lattice, an alternative meshing code was developed within the project. The code consists of an object-oriented Python framework in which the grids are built from macro objects in a block-structured manner. An example of a computed block-structured mesh of a spacer from the PSBT benchmark [92] is illustrated in Figure 2.8. The macro objects are discretized as blocks with an internal Cartesian discretization in  $(n_x, n_y, n_z)$  cells. The FVM formulation applied does not allow hanging nodes, and thus all intersecting blocks must have a consistent face discretization.

The block-structured method has the benefit of exact control of the discretization of each block and, accordingly, a shared preservation of all geometric regions between different meshes. However, there are drawbacks, and in particular the block-structured mesh will be governed by the regions of the finest mesh resolution which will propagate through the mesh. An example issue is seen in Figure 2.9, where the corners of the moderator mesh are influenced by the structure of the spacer, which gives artificial transitions between different regions in the mesh.

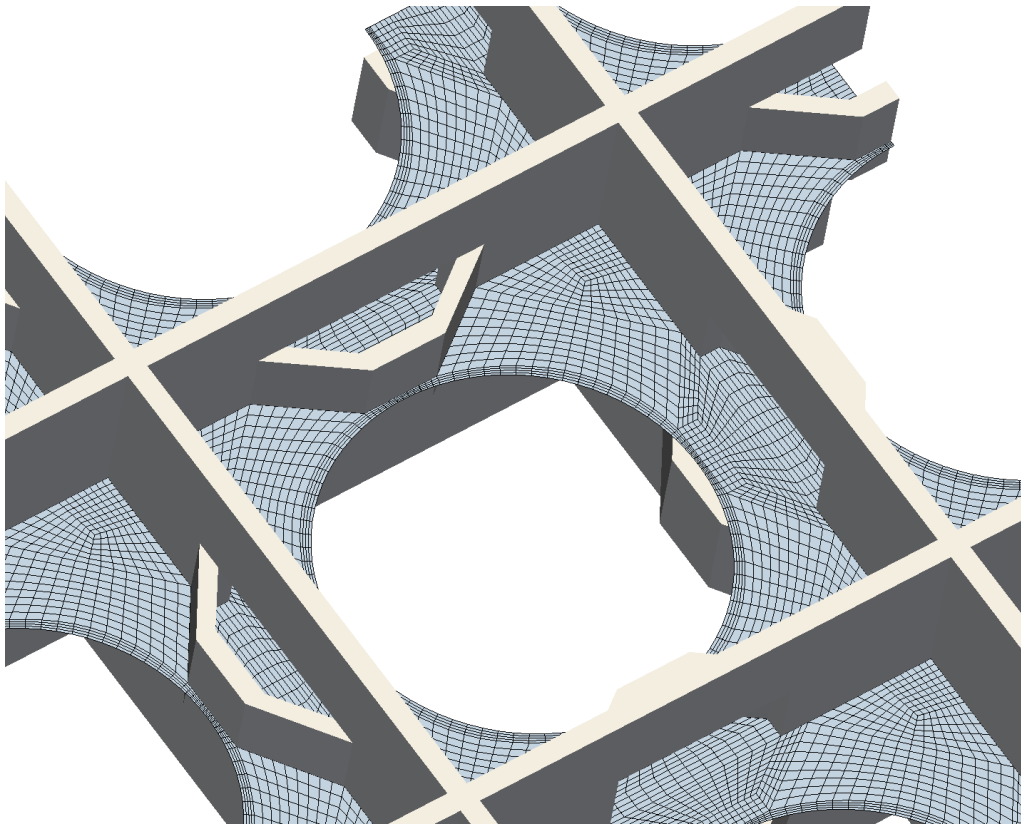




**Figure 2.7:** Example of mesh generated via SALOME (adaptation from [Paper I]).



**Figure 2.8:** Block-structured mesh of a spacer for a pin cell. The solid fuel pin is indicated by the low opacity orange region.



**Figure 2.9:** Example of moderator mesh for a pin cell including a spacer as shown in Figure 2.8.

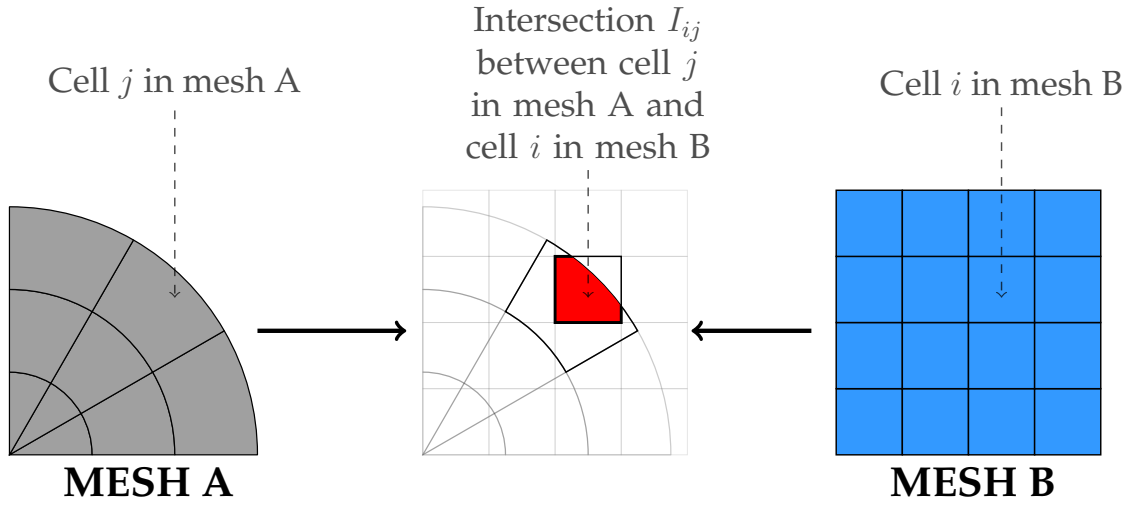
A configuration file ensures rapid control of the discretization and the system geometry (fuel pin radius, number of spacers, fuel pin lattice size, etc.). On the drawback side, it should be noted that the geometrical template structures are internally specified, e.g. the spacer geometry is hard coded from blueprints except for some general parameters such as height and material thickness. This is the price of the block-structured approach; the developer needs to more or less manually build the block structures internally. However, with the developed object oriented library, modifying the geometry or building new geometries are sufficiently fast. Also, this should be considered a programmatic approach to pure hexahedral meshing as can be done through interfaces in some mesh generators (such as e.g. ANSYS® ICEM [93]).

The developed mesh tool additionally creates the required initial and boundary conditions for all fields. Furthermore, the output of the mesh generator contains blocks with corresponding discretization, points and groups of block faces for the boundary conditions. The application generates the exemplified mesh (Figures 2.8 and 2.9) for 3 spacers in a 1 m long system in approximately 10 seconds. The short generation time is a benefit of the block-structured approach where not every single cell needs to be computed but only the blocks building the structures.

It should be noted that the generality of the tool is limited and, as discussed above, a more complicated spacer geometry would require an unstructured mesh. An extension of the tool would be to use unstructured meshes for certain regions of the multiphysics meshes, and specifically ensuring that the outer surfaces of such regions are consistent with neighboring block regions.

### 2.3.2 Mesh mapping

As described above, different meshes are applied for the different fields. To achieve data transfer between meshes of different resolution and structure a mapping algorithm is required. The problem can be solved by a point-to-point interpolation, advantageously accelerated by clever data sorting and decomposing the problem (see e.g. [94]). Alternatively, a direct overlap between the meshes could be computed such that volumetric intersections are used to map the data transfer. A schematic example of the latter approach is displayed for a 2D problem in Figure 2.10. An advantage of intersection approach is that the interpolation is directly conservative, and such a property was the rationale for choosing this approach for the current work. A more detailed discussion on the algorithm and its limitations is given in Paper IV.



**Figure 2.10:** Example of mapping of two overlapping meshes. The intersection  $I_{ij}$  is used to compute interpolation weights for two-way interpolation between cells  $i$  and  $j$ . [Paper IV]

### 2.3.3 Cross-section generation and utilization

Except for meshes and the material thermophysical data, the multiphysics tool requires macroscopic cross-sections for the neutron solvers. The actual application of the cross-sections in the neutronic equations is postponed to Section 3.1, but as the preparation of the cross-sections requires a methodology in itself, a description is given of the criteria for and how to efficiently compute such cross-sections.

#### Prerequisites and potential methodologies

Based on the objectives of sub-pin resolution in the fuel and the fine-mesh multiphysics coupling, the prerequisites are that the cross-sections should be:

- valid at a sub-fuel pin scale,
- discretized in  $G$  energy groups (where  $G \ll 100$ ),
- providing higher order ( $P_N$ ) scattering matrices,
- parametrized on the thermophysical state (temperature, density, etc), and
- useful in combination with the multiphysics setup as described in the previous section.

In particular, the last and the first point turns out to be challenging as the required scales are not the same as in the standard neutronics workflow (as described in Section 1.2.1). Three alternative approaches to generate the cross-sections were identified:

1. Generate macroscopic cross-sections directly from the nuclear data libraries with correct handling of resonances, with a large number of energy groups. This could either be done in a lower dimension spatial calculation or on parts of the actual simulated 3D domain. A detailed overview of the stages associated with lattice computations and cross-section generation is given by Knott and Yamamoto [3] and it is clear that this process would require a major effort in terms of methodology development, and it would later risk to significantly increase the simulation time.
2. Directly apply cross-sections generated by a lattice code. This option was investigated in the early stages of the project (primarily for the commercial codes CASMO-4E [95]), but it was found that the available output was provided on a scale much coarser than required. As described in Section 1.2.1, the lattice codes are aimed at generating fuel assembly cross-sections in few energy groups for the core solvers. As a result, the geometry and scales of the condensation and homogenization processes do not match the criteria for the fine-mesh simulations. It should be noted that there are examples of open source lattice codes (e.g. DRAGON [96]) which could potentially be modified or extended to produce the desired set of cross-sections. However, also for such alternatives there are severe limitations when it comes to non-standard geometries (e.g. a spacer) and again a major effort would be required to extend the code and develop such a model.
3. Compute macroscopic cross-sections with a Monte Carlo approach. This option has the benefit of a very flexible geometry, where most Monte Carlo codes supports combinations of a range of primitive mathematical geometries. The obstacle for this alternative is that macroscopic cross-section generation is not historically a standard functionality of such solvers. The long term state-of-the-art code MCNP [97] has been user modified for such generation [98], and the proposed methodology was tested. However, the implementation achieved was found inefficient and inconsistent wherefore the early attempts were abandoned. It should however be noted that the same group performed the reversed operation, i.e. performing calculations in MCNP based on macroscopic cross-section [99], which was successfully applied in noise calculations related to the current thesis [100]. As an alternative to MCNP, the code Serpent [101] was tested and found more suitable since macroscopic cross-section generation is a built-in utility and can be applied for specific regions of the simulated domain (such as a small part of a fuel pin).

Decidedly, all three alternatives have benefits and in particular the first option is tractable from a theoretical point of view. Under transient conditions, where the reactor is potentially far from critical, a general assumption of criticality might give significant impact on the cross-section [102]. With the first option, an on-the-fly approach could potentially compute the cross-sections valid for the actual, and not necessarily critical, state of the reactor. However, as noted above, such an

alternative is associated with a large development effort and a significant computational cost.

From a framework point of view, the third option was judged as the choice easiest to implement and deploy in the multiphysics solver since Serpent includes advanced geometries and cross-section generation and extraction. Nevertheless, the resolution required in this project is still not a default case and the code is not explicitly prepared for generation of many sets of cross-sections on small parts of a fuel assembly or fuel pin. For this reason, and to pre-process the cross-sections for use with the deterministic solver, a wrapper code was developed (described below).

Given the choice of Serpent it is tempting to entirely discard the idea of the deterministic neutron transport solver. Why should we use a discretized, condensed and homogenized approach to the neutron transport when we can readily perform simulations in continuous energy with no discretization errors associated using Monte Carlo? In particular, the availability of detailed geometrical descriptions and accurate solution to the transport problem are appealing. Although the arguments for a Monte Carlo approach are indeed strong for steady simulations, there are still some aspects that support the choice of a deterministic approach:

- *Transient simulations* – Whereas steady-state coupled reactor problems have been performed with Monte Carlo methods for a number of years, transient algorithms are less developed. Examples of simulations of short transients with Serpent (without delayed neutron handling) exist in the literature [103, 104]. Also methodologies for longer transients, i.e. where delayed neutron handling is essential have been proposed [105] but still seem to be under development and not yet sufficiently mature to be a viable option.
- *Simulation time* – Monte Carlo simulations are associated with a significant cost, in particular when compared to heavily optimized core simulators. In the current case of fine meshes the difference could be expected to be smaller but nevertheless significant.
- *Resolution of coupling* – In order to perform a direct fine-mesh coupling between the CFD and the Monte Carlo solvers, a fine geometrical resolution of the tallies would be required. Consequently, the Monte Carlo simulations must be run for a long time to reach low statistical errors for each small volume coupled to the fluid solver. To get a high precision for very small regions in space would induce a much higher cost than getting precise global parameters.
- *Inconsistent parallelization schemes* – Due to the independence between the simulated neutron histories the Monte Carlo software is most conveniently parallelized by duplicating the system, run on separate processors/threads and then recombining the data in a statistically consistent manner. In contrast, the large scale CFD solvers parallelize by domain decomposition. A tightly coupled parallelization schemes, such as presented in Figure 2.3,

would thus not be readily achievable. The monolithic code approach is particularly important for transient solutions with a lot of data exchanges.

Whereas the latter three points could be overcome by (extended) simulation times, the first point is decisive for the current project. In the long run, transient simulations are of particular interest and to get a relevant coupling to the CFD solver and to the scenario length of the transients, requires handling of the delayed neutrons.

### **Cross-section generation using Serpent**

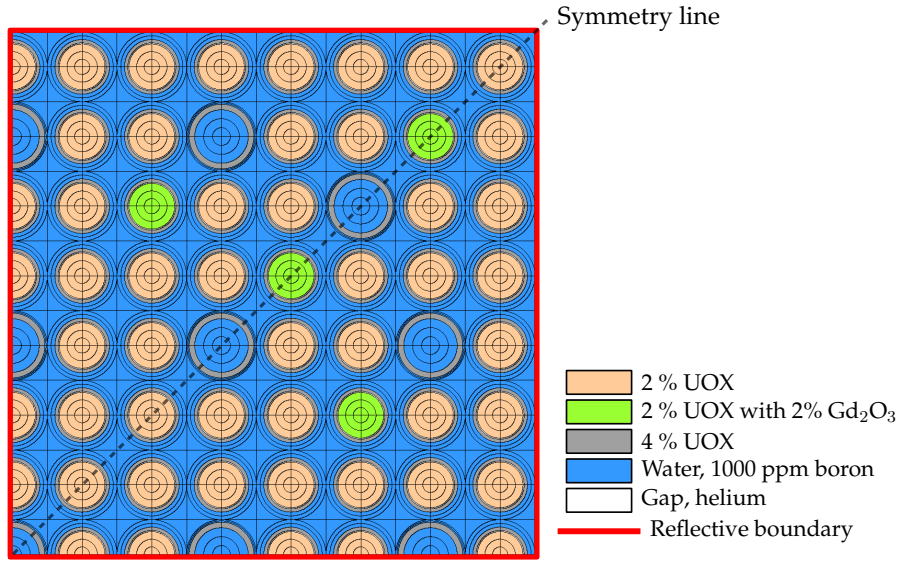
To generate cross-sections for the fine-mesh resolution in Serpent a wrapper application was developed. Similar to the mesh application described above, the tool is developed to automatically create a fuel assembly geometry, but here for the purpose of detailed cross-section generation. The application has the major benefit that the, quite error prone, input format of the Monte Carlo code is hidden. Instead a configuration file with much fewer options and only the necessary geometry details is sufficient to setup a case. The utility requires the following information:

- the lattice pitch together with the number of fuel pins, and the respective type of pin at each position,
- the desired resolution of the cross-section set in terms of radial and azimuthal regions in the fuel, gap, cladding and moderator as exemplified in Figure 2.11,
- the desired number of energy groups,
- the material properties, including densities and isotopic compositions, and
- the radial temperature profile in the fuel as well as in the moderator.

For the moderator, the application computes the relevant density properties and applies this to the material of the particular region in space associated with the specified temperature. It should be noted that Serpent handles the Doppler broadening associated with a certain material temperature, but the application still needs to determine and supply Serpent with the correct cross-section table.

In principle three dimensional sets of cross-sections could be computed with both a horizontal and axial resolution. However, as discussed above, finer resolutions require longer simulation times to get the desired statistical confidence intervals. Furthermore, as the cross-section generation is not at the core of the project, a reasonable assumption is to use axially independent cross-sections, i.e. relying on a horizontal slice with reflective boundary conditions in the axial direction.

Given the above specified information, an input file is created for Serpent and then run with the same tool. After the simulation is finished, the same application



**Figure 2.11:** Fuel pin discretization in horizontal plane, using 4 azimuthal and 8 radial regions per pin cell, in total 1775 regions [Paper IV].

computes temperature dependent cross-section files for the multiphysics and geometrical descriptions of each of the regions (radial and azimuthal information) in an OpenFOAM® specific format. The latter is necessary to compute sets of cells from the computational meshes for which each group of cross-sections should be applied.

The geometry in Figure 2.11 shows the actual cross-section regions, in total 1775 regions for the multiphysics computation in [Paper IV]. The Serpent input file for the presented case is more than 8000 lines long, which emphasizes the need for the automatic procedure. The input for the wrapper is 100 lines long.

Implicitly, the described process includes a number of assumptions whereof the most notable are that:

- the cross-section generation is performed under the assumption of periodic boundary conditions, i.e. a system with an infinite number of identical fuel assemblies, and
- the cross-sections are computed for static temperature profiles, a priori determined but potentially updated by an iterative process between the Serpent wrapper and the multiphysics simulations.

Similar to the meshing utility, the tool is implemented as an object-oriented library in Python. The object-orientation has many practical features when it comes to the geometry modeling. The simulation time of the script itself is negligible in comparison to the Monte Carlo simulation.



# Models for the coupled LWR single-phase problem

After the introduction of the multiphysics problem with current standard and high-fidelity approaches in Chapter 1 and the overview of the computational methods in Chapter 2, this chapter introduces the neutronic and thermal-hydraulic models. Except for describing the transport equations for each of the modules a brief discussion is given on the options available to solve the non-linear multiphysics problem.

On a side note, the step going from the mathematical formulation in this chapter to the implementation in the FVM framework described in the previous chapter is seemingly large. To provide enough detail to directly repeat the numerical experiments is unsatisfactorily difficult (and page consuming). Such a deep rift between reported equations and actual implementation is commonly seen in the open literature, and is unfortunate. Although the next chapters (4 and 5) introduce some additional details, the current chapter should be better seen as an overview description of the neutronics and thermal-hydraulics.

## 3.1 Formulation of the neutronic problem

The transport of neutrons in the reactor core is governed by the linear Boltzmann equation, here written in an integro-differential form, such that [106]

$$\frac{\partial}{\partial t}n(\mathbf{r}, \Omega, E, t) + \Omega \cdot \nabla \Psi(\mathbf{r}, \Omega, E, t) + \Sigma_T(\mathbf{r}, E, t)\Psi(\mathbf{r}, \Omega, E, t) =$$
$$\int_{(4\pi)} \int_0^\infty \Sigma_s(\mathbf{r}, \Omega' \rightarrow \Omega, E' \rightarrow E, t)\Psi(\mathbf{r}, \Omega', E', t)d\Omega'dE' +$$

$$\frac{\chi(E)}{4\pi} \int_0^\infty \nu \Sigma_f(\mathbf{r}, E', t) \Phi(\mathbf{r}, E', t) dE', \quad (3.1)$$

where the neutron density ( $n(\mathbf{r}, \Omega, E, t)$ ) is related to the angular neutron flux as

$$\Psi(\mathbf{r}, \Omega, E, t) = v(E) n(\mathbf{r}, \Omega, E, t), \quad (3.2)$$

which in turn is related to the scalar flux  $\Phi(\mathbf{r}, E, t)$  as

$$\Phi(\mathbf{r}, E, t) = \int_{(4\pi)} \Psi(\mathbf{r}, \Omega', E, t) d\Omega'. \quad (3.3)$$

The balance equation eq. (3.1) is not analytically solvable except for simplified problems and in general we need to rely on numerical methods to resolve the neutron distribution in the core.

In order to solve eq. (3.1) in the FVM framework as described above, we discretize the solution space consisting of time ( $t$ ), space ( $\mathbf{r}$ ), angle ( $\Omega$ ) and energy ( $E$ ). The space discretization was already briefly discussed in Section (2.2.1), and instead, the balance equation is first discretized in terms of neutron energy such a set of  $G$  energy intervals is defined as

$$[E_{\min}, E_{\max}] = \prod_{g=1}^G [E_g, E_{g-1}], \quad (3.4)$$

with the purpose of writing eq. (3.1) as a set of  $G$  discrete equations, coupled in the discrete energy space. Such a set of relations are computed by integrating the balance equation over each energy interval  $g$  with  $1 < g \leq G$ . As an example the scalar neutron flux  $\Phi(\mathbf{r}, E, t)$  is discretized as

$$\Phi_g(\mathbf{r}, t) = \int_{E_g}^{E_{g-1}} \Phi(\mathbf{r}, E, t) dE. \quad (3.5)$$

For the cross-sections and the other energy dependent parameters of eq. (3.1), flux averaged quantities need to be computed, where e.g. the energy discretized total cross-section is given by

$$\Sigma_{T,g}(\mathbf{r}) = \frac{\int_{E_g}^{E_{g-1}} \Sigma_T(\mathbf{r}, E) \Phi(\mathbf{r}, E) dE}{\int_{E_g}^{E_{g-1}} \Phi(\mathbf{r}, E) dE}. \quad (3.6)$$

The weighting procedure is a key aspect of the cross-section generation and due to the complex energy dependence (see the example of the fission cross-section in Figure 1.1), such weighting must be carefully and consistently performed. In the current work this is implicitly achieved by the internal flux weighting in Serpent. After discretizing all terms of eq. (3.1), the balance equation for energy group  $g$  is given by

$$\frac{1}{v_g} \frac{\partial}{\partial t} \Psi_g(\Omega) + \Omega \cdot \nabla \Psi_g(\Omega) + \Sigma_{T,g} \Psi_g(\Omega) = \int_{(4\pi)} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\Omega' \rightarrow \Omega) \Psi_{g'}(\Omega') d\Omega + \frac{\chi_g}{4\pi} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}, \quad (3.7)$$

where the space and time dependencies are left out for the sake of brevity and the neutron density  $n(\mathbf{r}, \Omega, E, t)$  is written in terms of the angular flux according to eq. (3.2).

For the time dependence, the thesis includes both steady-state and time-dependent neutronics solvers. For the steady-state solver the time dependence in eq. (3.8) is discarded and a normalization factor  $1/k_{\text{eff}}$  is multiplying the fission source of the balance equation such that

$$\Omega \cdot \nabla \Psi_g(\Omega) + \Sigma_{T,g} \Psi_g(\Omega) = \int_{(4\pi)} \sum_{g'=1}^G \Sigma_{s,g' \rightarrow g}(\Omega' \rightarrow \Omega) \Psi_{g'}(\Omega') d\Omega + \frac{\chi_g}{4\pi k_{\text{eff}}} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}, \quad g = 1, \dots, G. \quad (3.8)$$

The equation now takes the form of an eigenvalue problem where the eigenvectors are the (angular) neutron fluxes and the smallest eigenvalue corresponds to the inverse of the criticality factor ( $k_{\text{eff}}$ ).

For the transient approach the time derivative is retained and in addition the fission source of neutrons is split in two parts, the contributions from the prompt neutrons and the delayed neutrons. The prompt neutrons are released immediately after the fission event. In contrast, the delayed neutrons are released after decay of the fission products with a varying time delay. In order to account for the accumulation of such precursors of delayed neutrons and the contribution to the balance equation, additional conservation equations for the precursors are added. Due to the large number of different fission products decaying, the precursors are routinely grouped in  $I$  groups, with one conservation equation per group. The balance equation for the transient problem is then given by

$$\begin{aligned} \frac{1}{v_g} \frac{\partial \Phi_g(\mathbf{r}, t)}{\partial t} \nabla \cdot \mathbf{J}_g(\mathbf{r}, t) = & -\Sigma_{T,g}(\mathbf{r}, t) \Phi_g(\mathbf{r}, t) + \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) \\ & + (1 - \beta) \chi_g^p \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) + \chi_g^d \sum_{i=1}^I \lambda_i C_i(\mathbf{r}, t), \quad g = 1, \dots, G, \end{aligned} \quad (3.9)$$

with the precursor concentrations ( $C_i$ ) calculated as

$$\frac{dC_i(\mathbf{r}, t)}{dt} = \beta_i \sum_{g'=1}^G \nu_{g'}(\mathbf{r}, t) \Sigma_{f,g'}(\mathbf{r}, t) \Phi_{g'}(\mathbf{r}, t) - \lambda_i C_i(\mathbf{r}, t), \quad i = 1, \dots, I. \quad (3.10)$$

As seen from eqs. (3.9) and (3.10) the equations are interdependent and must be solved together, either by explicit iteration or in a more implicit manner.

For the angular discretization, two different alternatives will be considered in this thesis, namely the diffusion approximation, in which the angular dependencies are removed, and the discrete ordinates method.

### 3.1.1 Diffusion approximation

The diffusion approximation for the angular dependent neutron transport equation is achieved in three steps. First, the neutron transport equation is integrated over all angular space such that eq. (3.8) gives

$$\nabla \cdot \mathbf{J}_g + \Sigma_{T,g} \Phi_g = \int_{(4\pi)} \int_{(4\pi)} \sum_{g'=1}^G \Sigma_{s,g}(\Omega' \rightarrow \Omega) \Psi_g(\Omega') d\Omega d\Omega' + \frac{\chi_g}{k_{\text{eff}}} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Phi_{g'}. \quad (3.11)$$

Secondly, the scattering kernel ( $\Sigma_{s,g}(\Omega' \rightarrow \Omega)$ ) is assumed to be isotropic, i.e.

$$\Sigma_s(\mathbf{r}, \Omega' \rightarrow \Omega, E' \rightarrow E) = \frac{\Sigma_{s0}(\mathbf{r}, E' \rightarrow E)}{4\pi}. \quad (3.12)$$

Finally, we apply Fick's law to approximate the current as [107]

$$\mathbf{J}(\mathbf{r}, E) \approx -D(\mathbf{r}, E) \nabla \Phi(\mathbf{r}, E). \quad (3.13)$$

Inserting eqs. (3.12) and (3.13) in eq. (3.11) then gives the final form of the steady-state diffusion equation as

$$-\nabla \cdot (D_g \nabla \Phi_g) + \Sigma_{T,g} \Phi_g = \sum_{g'=1}^G \Sigma_{s0,g' \rightarrow g} \Phi_{g'} + \frac{\chi_g^p}{k_{\text{eff}}} \sum_{g'=1}^G \nu \Sigma_{f,g'} \Phi_{g'}. \quad (3.14)$$

Advantageously, the diffusion equation efficiently reduces the solution space for the neutron distribution as only one equation is solved per energy group. However, the approximation of isotropic scattering and the use of Fick's law is known to reduce the accuracy, in particular for heterogeneous regions such as close to the interface between the fuel pins and the moderator. A more rigorous derivation of the diffusion equation using  $P_1$  theory is given in Bell and Glasstone [107].

### 3.1.2 Discrete ordinates method

As an alternative to the diffusion approximation of the angular flux, there are many methods to actually resolve the angular flux, or at least some degree of it. Just like space or time, the angular dimension requires discretization. In the spherical harmonics ( $P_N$ ) method the angular flux is expanded in terms of a spherical harmonics base of order  $N$  (with couplings to higher order terms neglected), which results in a set of coupled equations for the flux expansions (see e.g. [68, 106]). In the discrete ordinates method ( $S_N$ ) the angular space is instead covered with a set of discrete directions. The neutron transport equation is solved for each such direction. The latter method ( $S_N$ ) is the method applied in this thesis.

In more specific terms, for the steady state discrete energy problem, eq. (3.8) is written for one specific ordinate (i.e. streaming direction)  $\Omega_m$  [108]

$$\Omega_m \cdot \nabla \Psi_{m,g} + \Sigma_{T,g} \Psi_{m,g} = S_{m,g} + \frac{1}{k} F_{m,g}, \quad (3.15)$$

where the fission source ( $F_{m,g}$ ) is defined as

$$F_{m,g} \equiv \chi_g \sum_{m'}^M w_{m'} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'} \Psi_{m',g'}, \quad (3.16)$$

and the scattering term is expanded in terms of Legendre polynomials ( $P_l$ ) and written for a discrete number of ordinates  $M$  such that

$$S_{m,g} \equiv \sum_{l=0}^L (2l+1) \sum_{m'}^M P_l(\Omega_m \cdot \Omega_{m'}) w_{m'} \sum_{g'=1}^G \Sigma_{s,l,g' \rightarrow g} \Psi_{m',g'}. \quad (3.17)$$

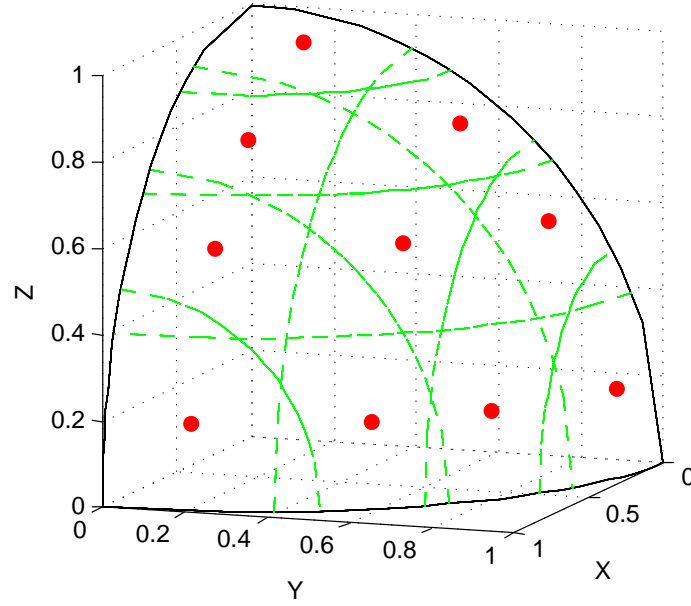
The scalar flux ( $\Phi_g$ ) is computed as a weighted sum of the flux for each direction, such that

$$\Phi_g = 4\pi \sum_m^M w_m \Psi_{m,g}, \quad (3.18)$$

where the weights ( $w_m$ ) are associated with the chosen quadrature set as discussed next.

The set of discrete directions ( $\Omega_m$ ) and weights ( $w_m$ ) is decisive for the accuracy of the solution of the  $M$  coupled transport equations. The optimal set is unfortunately problem specific (see e.g. [110]). However, in general a larger number of discrete directions will recover the angular flux with a better accuracy. In this thesis specifically, the level symmetric quadrature set is applied (for further details see e.g. Hébert [68]). An example of the set of directions for such a set of order 9 is shown for the first octant in Figure 3.1. As seen in the figure, the set of directions is symmetric as regards all Cartesian axes.

Computationally, the most costly part of eq. (3.15) is the evaluation of the scattering term, where each evaluation of  $S_{m,g}$  for a specific  $m$  and  $g$  requires



**Figure 3.1:** Example discretization of an octant of the angular space using level symmetric quadrature of order 8 with  $\mu_1 = 0.20$  according to [109].

$(L \times M \times G)$  summations. In order to reduce the cost of this computation the scattering source is projected on a spherical harmonics basis, specifically the real spherical harmonic functions as defined by Hébert [68], such that eq. (3.17) can be written

$$S_{m,g} = \sum_{g'=1}^G \sum_{l=0}^L (2l+1) \sum_{r=-l}^l R_{lr} \phi_{g,l,r} S_{s,l,g' \rightarrow g}. \quad (3.19)$$

The quadrature set applied in this thesis represents the spherical harmonics orthogonality up to a degree  $L = N/2$  [68] and therefore a smaller number of expansion coefficients  $\phi_{g,l,r}$  than the actual number of discrete ordinates are required.

The  $S_N$  method is well known to converge slowly for many types of problems [111]. Commonly, different accelerating schemes are applied to decrease the number of iterations required to resolve the angular interdependence between the different directions in the scattering source (explicitly in eq. (3.17) or implicitly via the projection on the spherical harmonics in eq. (3.19)). One classical method is the diffusion synthetic acceleration (DSA) [112]. Successful implementations of such accelerated schemes are known to be dependent on a discretization consistent with the original  $S_N$  implementation [113]. Moreover, some classical schemes like the DSA have been shown to work worse in cases of multidimensional cases with strong material heterogeneities (which very well describe the systems of interest in this thesis). Another possible approach, is to solve the problem using a Krylov method, which has been reported to decrease the dependence on the discretization [113]. What regards the neutronics solver developed for this thesis, a Krylov method is by hypothesis a better candidate for the unstructured (or at least non-Cartesian) approach. However, in the current scope, the coupling

between the ordinates was resolved by source iteration.

## 3.2 Single-phase fluid flow and heat transfer

The single-phase flow problem in a LWR is governed by the conservation equations for mass, momentum (the so called Navier-Stokes equations) and enthalpy written as [114]

$$\frac{\partial \rho(\mathbf{r}, t)}{\partial t} + \nabla \cdot (\rho \mathbf{u})(\mathbf{r}, t) = 0, \quad (3.20)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t}(\mathbf{r}, t) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u})(\mathbf{r}, t) = \nabla \cdot \tau(\mathbf{r}, t) - \nabla p(\mathbf{r}, t) + \rho(\mathbf{r}, t)g, \quad (3.21)$$

and

$$\frac{\partial(\rho h)}{\partial t}(\mathbf{r}, t) + \nabla \cdot (\rho \mathbf{u} h)(\mathbf{r}, t) = -\nabla \cdot q''(\mathbf{r}, t) + q'''(\mathbf{r}, t) + \bar{\tau}(\mathbf{r}, t) : \nabla \otimes \mathbf{u}(\mathbf{r}, t) + \nabla \cdot (\mathbf{u} p)(\mathbf{r}, t). \quad (3.22)$$

The interpretation of the heat source terms ( $q''$  and  $q'''$ ) is discussed in detail in Section 3.2.3. The complexity of the equations is manifold. First, eqs. (3.20)–(3.22) are all interdependent through the fluid velocity ( $\mathbf{u}$ ) and (except for the continuity equation) through the pressure ( $p$ ). Secondly, the equations are non-linear in the solution variables  $\mathbf{u}$ ,  $h$  and  $p$  (or alternatively  $\rho$ ). Thirdly, all material data in eqs. (3.20)–(3.22) vary with the thermophysical state of the system, which in turn will change due to the enthalpy rise in the heated channels in the core. For all three mentioned reasons, an iterative (or non-linear) algorithm is required to resolve the dependencies, and this is further discussed in Section 3.2.2 below.

The single-phase problem with low Mach numbers can often be solved as an incompressible flow, for which the criteria on the density is that

$$\frac{1}{\rho} \frac{D\rho}{Dt} = 0. \quad (3.23)$$

For the single-phase reactor core problem, the characteristic velocity is indeed significantly lower than the speed of sound in the coolant ( $Ma < 0.01$ ). However, due to the heating of the water the density change will violate the condition in eq. (3.23). Considering the thermodynamic equation of state we can write [114]

$$\frac{1}{\rho} \frac{D\rho}{Dt} = \alpha \frac{Dp}{Dt} + \beta \frac{DT}{Dt}, \quad (3.24)$$

where  $\alpha$  is the isothermal compressibility and  $\beta$  is the bulk thermal expansion coefficient. Although  $\alpha$  is small in the case of liquid water, the flow should still be considered as low speed compressible due to the changes caused by the temperature change.

For the incompressible flow, the coupled eqs. (3.20)–(3.21) are typically solved in terms of pressure and velocity, whereas a compressible solver is often posed

in terms of velocity and density (see e.g. [65, 115]). In the current case, a so-called weakly compressible approach is needed, with special attention given to the dependence on temperature for all transport coefficients.

In addition to the fluid conservation equations, an energy equation for the fuel pins must be formulated. Such a balance equation is coupled to the energy equation of the water at the interface of the fuel pin and accordingly results in a conjugate heat transfer problem between the fuel pins and the fluid. The treatment of the heat transfer is discussed in detail in Section 3.2.3.

### 3.2.1 Turbulence

The conservation equations (3.20)–(3.22) describe local and instantaneous flow and due to the large range of time and length scales involved a direct solution to industrial scale problems is still not feasible. Indeed, solutions resolving all the characteristic scales of the flow, i.e. to the smallest dissipation scales of the turbulence, can be computed with Direct Numerical Simulation (DNS) (see e.g. [116]). However, due to the enormous computational requirements only limited domains with relatively low Reynolds numbers can be simulated even with the current state-of-art HPC resources.

Instead of the direct approach, we need to approach the equations with a filtered or averaged approach. From a computational perspective, two methodologies, or rather classes of methodologies, are commonly found in CFD, namely Reynolds-Averaged Navier Stokes (RANS) and Large Eddy Simulations (LES), which are both popular and widely applied in nuclear engineering [117]. In the current thesis the prior has been used as a first approach. Later experimentation with LES was performed and due to the generally increased resolution, the cost of such simulations are significantly larger than the RANS counterpart. Nevertheless, LES is a future candidate for the coupled transient calculations within the multiphysics tool.

#### RANS

In RANS, a filtering of the conservation equations is performed via a decomposition of each field in its mean value and a temporally fluctuating component. The details of the procedure is well covered in the literature (see e.g. [65, 116, 118, 119]), and here we will be satisfied with the final form of the mass, momentum and enthalpy equations after the insertion of the decomposed velocity, pressure, and enthalpies and time averaging the equations given by

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{U}) = 0, \quad (3.25)$$

$$\frac{\partial (\rho \mathbf{U})}{\partial t} + \nabla \cdot (\rho \mathbf{U} \otimes \mathbf{U}) = \nabla \cdot \boldsymbol{\tau} - \nabla \cdot \overline{\rho \mathbf{u}' \otimes \mathbf{u}'} - \nabla P + \rho g, \quad (3.26)$$



and

$$\frac{\partial(\rho H)}{\partial t} + \nabla \cdot (\rho \mathbf{U} H) = -\nabla \cdot \overline{q''} + \overline{q'''} + \nabla \cdot (\mathbf{U} P) + \overline{\nabla \cdot (\mathbf{u}' p')} + \tau \nabla \otimes \mathbf{U} - \overline{\nabla \cdot (\rho \mathbf{u}' h')}, \quad (3.27)$$

where space and time dependence  $(\mathbf{r}, t)$  is omitted for brevity, capital letters are used for mean quantities  $(\mathbf{U}, H, P)$  and primes indicate the fluctuating components  $(\mathbf{u}', h', p')$ .

To close eqs. (3.25)–(3.27), the terms with fluctuating components must be modeled. As regards the momentum equations, the Reynolds stress tensor  $\mathbf{u}' \otimes \mathbf{u}'$ , could be solved for by using six additional (all permutations of  $u'_i u'_j$ , assuming symmetry) equations which then model the Reynolds Stresses. Alternatively, and computationally cheaper, the Boussinesq assumption could be applied such that in a tensor notation

$$\overline{\rho u'_i u'_j} = \mu_t \left( U_{i,j} + U_{j,i} - \frac{2}{3} U_{k,k} \delta_{i,j} \right) - \frac{2}{3} \rho k \delta_{i,j}, \quad (3.28)$$

where the model is posed in terms of a turbulent kinetic viscosity  $\mu_t$  and the turbulent kinetic energy  $k$ . Instead of solving for the Reynolds stresses, an equation for  $k$  is solved, and in turn additional equations could be added to model some source term in the  $k$ -equation. The open literature contains a plethora of different models to close the equation for turbulent kinetic energy with different supposed strengths and weaknesses. In this thesis the standard  $k - \epsilon$  model is used, i.e. a two-equation model to compute the turbulent kinetic energy and the turbulent dissipation combined into a model for the turbulent kinetic viscosity.

In addition to the closure for the Reynolds stresses, models are required for terms with primed quantities in the enthalpy equation. As regards the last term of eq. (3.27), this can be modeled as an additional contribution to the thermal conductivity in the spirit of the Boussinesq assumption (covered in Section 3.2.3) whereas the term  $\overline{\nabla \cdot (\mathbf{u}' p')}$  is assumed negligible.

It should be noted that the equations above are in principle the Unsteady RANS equations (URANS) as the time derivatives are kept. For the steady-state version of the model (RANS) the time-derivatives are taken out, and this is the model applied in Papers I, II and IV. The interpretation of URANS is not without controversy (see e.g. [117]) and the fluctuations in the results computed with such a model are likely doubtful at the best.

### 3.2.2 Pressure and velocity algorithms

As noted above, eqs. (3.20)–(3.22) are coupled via pressure and velocity and an iterative or non-linear approach is required to resolve the dependencies. The most classical approaches in CFD is to solve the problem linearized and one component and equation at a time, i.e. using a so-called segregated approach. The SIMPLE algorithm [81] has been the de facto standard for steady-state pressure-based

incompressible solvers since the 1970s. An important element of the method is the formulation of a pressure equation given the continuity and Navier-Stokes equations. Interestingly, the mass (or continuity) equation lacks an explicit dependence on pressure. In a mathematical formalism, eqs. (3.20) and (3.21) written in a matrix format give a saddle-point problem, essentially lacking the diagonal term in the system matrix (see e.g. [120]). The solution proposed in SIMPLE could formally be seen as a two stage procedure where the velocity is approximated from the previous pressure solution, followed by the solution of the pressure field computed via the Schur complement of the system matrix.

In a similar manner as the SIMPLE algorithm, the unsteady PISO algorithm [82] relies on an iterative approach to resolve the pressure and velocity dependencies (for further details on the formulation and the implementation see e.g. [80]). The SIMPLE algorithm was applied in all steady-state simulations in this thesis (Papers I, II and IV) and PISO for the transient simulations (Papers V and VIII).

### 3.2.3 Heat transfer problem

The LWR core requires the conjugate heat transfer problem to be solved as the heat generated in the fuel pins is extracted via the fluid coolant. The enthalpy balance in the fluid was already described by eq. (3.22). However in order to directly solve both the fluid and the solid regions in a monolithic, implicit system it is convenient to use the same solution variable for energy throughout the domain. As the solid region is described by the temperature conduction equation

$$\rho(\mathbf{r}, T, t)c_p(\mathbf{r}, T, t)\frac{\partial T(\mathbf{r}, t)}{\partial t} = \nabla \cdot (K(T)\nabla T(\mathbf{r}, t)) + q'''(\mathbf{r}, t), \quad (3.29)$$

the fluid enthalpy equation is re-written in terms of temperature [109]. The resulting fluid equation is written as

$$\frac{\partial(\rho c_p T)}{\partial t} + \rho c_p \vec{U} \cdot \nabla T = \beta \vec{U} \cdot \nabla P + \nabla \cdot (K \nabla T) + q''', \quad (3.30)$$

where  $\beta$  is the thermal expansion coefficient of the fluid and where Fourier's law of conduction has been applied to model the surface heat flux in terms of a diffusion term. The thermal conductivity parameter ( $K$ ) additionally includes a contribution from the turbulent diffusivity.

Although the volumetric source term  $q'''$  is kept in eq. (3.30), no actual sources in the liquid are introduced. In principle the term could have introduced some source terms due to capture of gamma rays [3], released in the fission events. However, no transport of the gamma is covered in this thesis. In the temperature equation for the fuel, the volumetric source term is computed from the recoverable energy of the fissions and is thus the explicit coupling from the neutronics to the thermal-hydraulics.

In contrast to the pressure-velocity coupling described in Section 3.2.2, the temperature equations require no special procedures in the discretization or equation reformulation. Instead a potential issue arises from the many solid regions which are separately connected to the fluid region. To solve the entire heat transfer problem all fuel pins and the fluid region need to be simultaneously converged. For iterative methods, such as the Neumann-Dirichlet method [121], or in a more general form non-overlapping Schwarz decomposition methods [122], the many material regions risk to give a very slow iterative problem. As an alternative, all regions could be formulated in a combined system of equations and consequently solved in a concurrent and fully implicit manner. The latter approach is applied in this thesis and in practice realized by coupling the temperature at the boundaries implicitly via a harmonic interpolation that preserves the conservation over the boundary faces between the solid and fluid regions.

In theory, the heat transfer by thermal radiation is required, at least for the gap and gas inside it which is not opaque to the radiation. As a first approximation, a frequency independent radiative heat transfer equation could be applied (see e.g. [123]). The black-body source term for the radiation would then be computed according to the Stefan-Boltzmann law such that,

$$E_b(\mathbf{r}, T) = \sigma_{SB} n^2 T^4(\mathbf{r}), \quad (3.31)$$

where  $\sigma_{SB}$  and  $n$  correspond to the Stefan-Boltzmann constant and the refractive index of the medium, respectively. Due to the large exponent on the temperature, the thermal heat radiation is either not significant (low temperatures) or completely dominating the heat transfer (high temperatures). A thermal radiation model for the gap was included in Paper I. The effect of radiation was studied in more detail in a previous work [124], where it was found that, for nominal PWR conditions, the heat transfer due to radiation in the gap was insignificant. For Paper IV the heat transfer in the gap was therefore modeled similarly to the solid regions, i.e. as dominated by conduction.

## 3.3 Multiphysics formulation and algorithms

After the formulation of the separate neutronic and thermal-hydraulic models, the attention is turned to the multiphysics and the algorithms to solve the combined problem of neutron transport, fluid continuity and momentum conservation and the conjugate heat transfer between the fluid and the solid fuel pins.

As touched upon in the introduction (Section 1.3.2), the first choice altogether is whether to solve the coupling in a single code or by application of multiple tools. Some of the respective benefits and drawbacks of the approaches were mentioned in Section 2.1.3, and in particular the potential excessive computational cost of parallelizing the segregated approach was discussed. As outlined in Section 2.2, this thesis is based on a single code approach.

To understand the complete multiphysics algorithm, it should be emphasized that there are many layers to the problem that need to be addressed, namely:

- *Multiphysics* – The thermal-hydraulic equations rely explicitly on the power density computed from the neutronics via the source term in the fuel temperature eq. (3.29), whereas the neutronics implicitly relies on the thermophysical state via the density and temperature dependence of the cross-sections.
- *Non-linearities* – As discussed in Section 3.2 the conservation equations for the fluid are all coupled and non-linear in the solution variables. Also the steady-state neutronic problem is non-linear due to the criticality factor.
- *Implicit dependencies* – Both the neutronics and the thermal-hydraulics rely on material parameters that are dependent on the thermophysical state and such dependencies are not expressed in algebraic relations, rather computed in a black box manner, i.e. relying on e.g. a database of values.

From the above list it is immediately clear that the system cannot be directly described with a linear equation system (i.e.  $Ax = b$ ) rather it is described in a non-linear manner such that in a generic notation we need to solve

$$F(x) = 0, \quad (3.32)$$

where  $x$  would be a solution vector of all unknowns ( $\Phi_g/\Psi_{m,g}$ ,  $\mathbf{U}$ ,  $T$ ,  $P$  in the steady-state case and with turbulence excluded).

In principle a problem like eq. (3.32) could be solved using a non-linear solution method like Newton's method, by directly computing the inverse of the Jacobian of the system of equations such that for an iterate  $m$  of the solution variable  $x$  we have

$$J(x_m)(x_{m+1} - x_m) = F(x_m), \quad (3.33)$$

where the solution of the system would give the next iterate of the solution variable  $x_{m+1}$ . To avoid the inverse of the Jacobian, eq. (3.33) can be solved by a standard linear solver. However, to even formulate the Jacobian might be both expensive and difficult. Due to the use of "black-boxes" in terms of the thermophysical parameters and cross-sections, a direct analytical form of the Jacobian is in principle impossible, and for all practical methods based on eq. (3.33) a numerical estimate of  $J$  is instead computed.

In the other end of the scale, a fully explicit approach is to solve the system with Picard iterations, where each separate module of physics is solved for constant values of all other equations. Considering the steady-state problem eq. (3.32) could be formulated in terms of the above fluid conservation equations and neutron transport equation as

$$F \begin{pmatrix} \phi \\ P \\ \mathbf{U} \\ T \end{pmatrix} = \begin{pmatrix} F_\phi(\phi, p, T) \\ F_P(\mathbf{U}, T) \\ F_{\mathbf{U}}(P, T) \\ F_T(\mathbf{U}, P, \Phi) \end{pmatrix} = \begin{pmatrix} M(T, p)\phi - \frac{1}{k_{\text{eff}}}F(T, p)\phi \\ H_P(\mathbf{U}, T)P = S_P \\ H_{\mathbf{U}}(P, T)\mathbf{U} = S_{\mathbf{U}} \\ H_T(\mathbf{U}, T)T = S_T(\mathbf{U}, P, \Phi) \end{pmatrix} = \mathbf{0}, \quad (3.34)$$

where the operators  $M$  and  $F$  are determined from the neutron diffusion equation (eq. 3.14) or the  $S_N$  equation (eq. 3.15), and  $H$  and  $S$  are determined from the mass conservation (eq. 3.25, formulated as a pressure eq. via SIMLPE or PISO), the momentum conservation (eq. 3.26) and the temperature equations for the solid (eq. 3.29) and the fluid (eq. 3.30), respectively. It should be noted that the operators  $M$  and  $F$  are also dependent on the criticality factor ( $k_{\text{eff}}$ ), which could in principle be added to the state vector above. Although the exact definition of each of the operators is not important for the reasoning, it should be noted that each such operator in itself corresponds to a non-linear problem, i.e. even when considering all other variables fixed.

A Picard iteration would, with the formulation as in eq. (3.34), be computed by solving each of the four equations with the other parameters frozen. In terms of coding effort such an approach is simple and the respective algorithms for solving the neutronics and thermal-hydraulics could be kept without modification. For example, the iteration between the pressure and velocity as performed in the SIMPLE algorithm could be applied unchanged. The direct use of Picard iterations has been shown to converge slowly for some multiphysics problems (as discussed in [125]). However, the method avoids formulating the actual Jacobian and thus a higher number of iterations could be accepted due to the lower cost per iteration.

Recently, the use of non-linear formulations has drawn a lot of attention in the multiphysics communities, including formulations such as the Jacobian Free Newton-Krylov method (JFNK) (see e.g. [126] and applied in [127]) and Anderson Mixing (originally in [128] and e.g. applied to reactor multiphysics by [129]). The former is known to require sufficiently good preconditioning [127], for example realized by a Picard style inner iteration. The latter could be seen as a direct acceleration of the fixed point iterations and constitute an interesting choice as the modifications required are small.

It should be noted that in principle two layers of non-linear solvers could be applied to solve eq. (3.34). Besides solving the outer, multiphysics iteration in a non-linear fashion, a non-linear approach could be applied for the separate equations, which has been done for neutronics [130, 131]. Also for CFD the implicit approaches have gained interest (see e.g. [132]) although the segregated fixed point algorithms like SIMPLE and PISO still prevail.

In the current thesis, the multiphysics couplings and the respective physics modules are all solved in a Picard iteration manner. The primary reason for this was the simplicity of the method and that the performance of the multiphysics iterations was judged acceptable in the early tests of the solver as discussed further in the results. The use of acceleration by e.g. Anderson mixing is interesting and should be a candidate for further studies on the coupled CFD/neutronics problem. Notably, many of the referenced examples of application of e.g. JFNK are targeted at macroscopic models, on the scale of the system codes, whereas the current thesis is focused on coupling on much smaller scales. The extent to which

this difference would manifest in better or worse acceleration with the mentioned methods is not clear and thus it is open for future investigations.

# Steady-state coupled solver application and analysis

In this chapter the outlined computational methodology and the defined physical models are applied together to simulate fine-mesh multiphysics for PWR like conditions. The chapter includes the results from Papers I, II and IV. Paper I reports the first version of the code applied to a system of  $5 \times 5$  fuel pins with a checkerboard pattern of low and high enrichment fuel pins. Whereas Paper I is entirely based on a diffusion solver for the neutronics, Paper II presents an implementation of  $S_N$  and accompanying results that compare the flux profiles from diffusion and discrete ordinates. Paper IV is again focused on the overall behavior of the framework and the code presented corresponds to a re-write as compared to Paper I.

Many of the presented results are to some extent a proof of principle and an attempt to present the feasibility of highly-resolved multiphysics simulations of LWR single-phase systems. As examples of this, the papers include figures for the computational effort and the relative cost of each of the modules in combination with the number of iterations required to solve the steady-state problem. Nevertheless there are results which are of physical interest, such as the results of the fully resolved simulations compared to some averaged and non-resolved simulations.

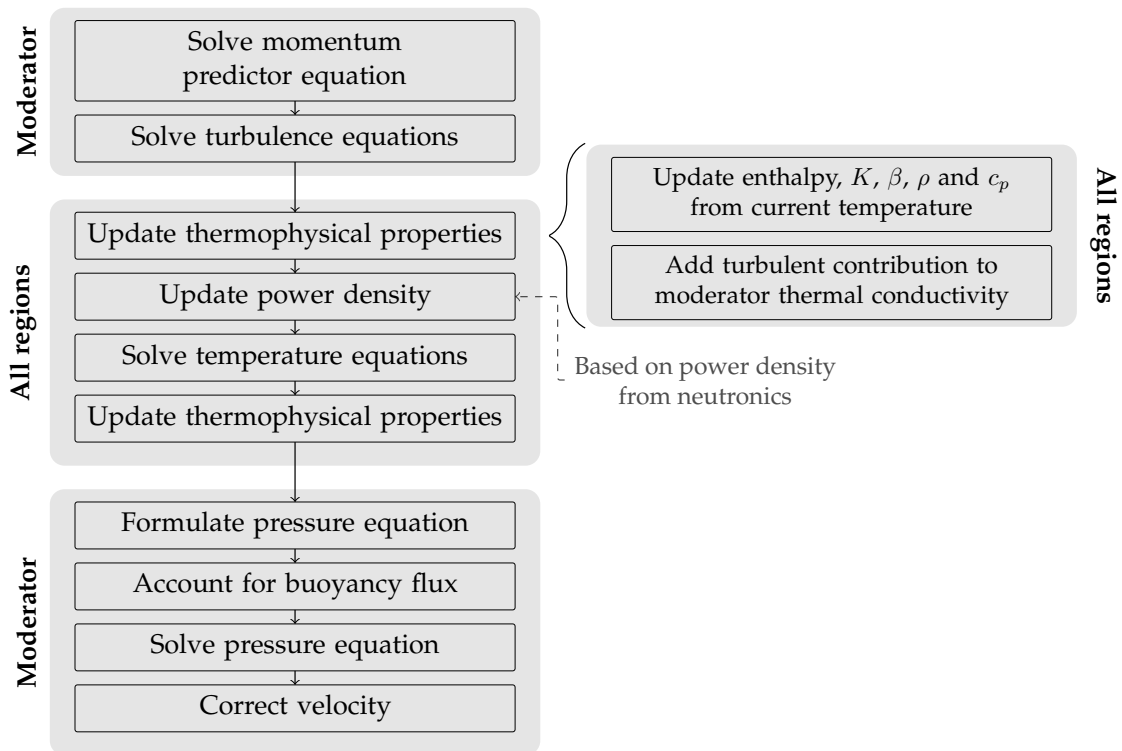
## 4.1 Implementation and framework details

Although many of the general aspects of the implementation of the multiphysics FVM code were described already in Chapter 2, some additional details specific to the steady-state solver are presented here. The framework was re-written between Paper I and Paper II. One of the major differences between the different

implementations was that whereas the first version used the same mesh for all fields of physics, multiple different meshes were supported in the re-written version. The single mesh approach was early judged as a drawback and the version including the mesh-to-mesh transfer capability (see Section 2.3.2), resulted in a more flexible code. In this summary, the description covers only the methodology corresponding to the second implementation.

#### 4.1.1 SIMPLE algorithm and heat transfer

The steady-state single phase solver relies on the briefly described SIMPLE algorithm (see Section 3.2.2). The steps of the algorithm are schematically outlined in Figure 4.1. As seen from the algorithm, the moderator momentum predictor equations and the turbulence are solved prior to the CHT problem, where the latter treated with the described monolithic approach for the temperature equation.



**Figure 4.1:** Thermal-hydraulics solver methodology.

As regards the stability of the scheme, the SIMPLE approach typically requires some under-relaxation and for the iterative algorithm presented in Figure 4.1, the pressure and velocity under-relaxations were typically 0.3 and 0.7, respectively. For the temperature and the turbulence equations less under-relaxation was generally needed.



### 4.1.2 Discrete ordinates solver

The algorithm of the discrete ordinates and eigenvalue solver, applied in Paper II and Paper IV, is outlined in Figure 4.2. The approach follows a standard scheme of an iterative approach to resolve the group and ordinate interdependencies.

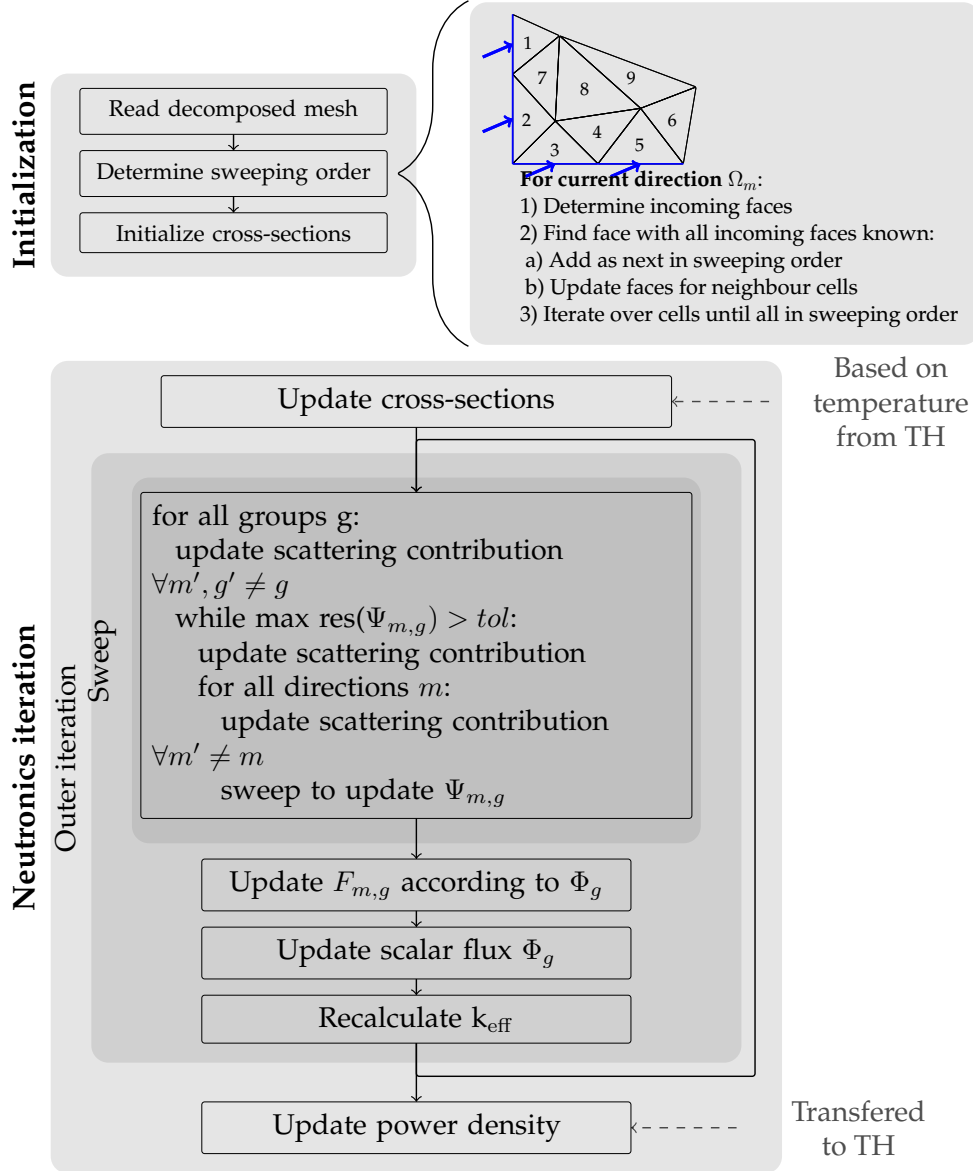


Figure 4.2: Applied algorithm for the discrete ordinates method. [Paper IV]

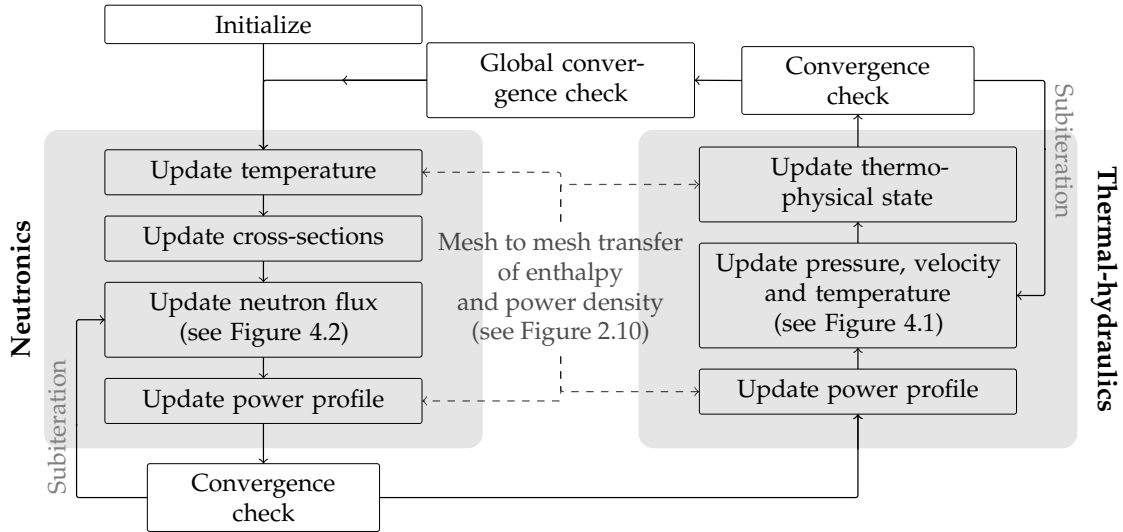
The solver is applied to the generally unstructured neutronics mesh which requires some additional attention. In the reactor context, discrete ordinates have often been implemented only for Cartesian grids, although late high-fidelity exceptions discretized with FEM are notable (see e.g. [133]). Unstructured implementations of  $S_N$  with FVM have for a long time been applied for radiative heat transfer (see e.g. [134]) with a step scheme discretization of the  $S_N$  equations. The

latter is in principle identical to an upwind differencing scheme as applied for the convective terms in the fluid momentum equation. Although the step scheme is accurate to first order only, it is stable and easy to implement and has therefore been the choice for this thesis.

As indicated in Figure 4.2, the  $S_N$  equations are solved using a pre-calculated sweep order. Since the only spatial dependence coupling with neighbor cells enters in the streaming term, each given direction ( $\Omega_m$ ) can be swept in an order such that all cells only need to be updated once. The algorithm applied is based on the work by Plimpton, Hendrickson, Burns, McLendon, and Rauchwerger [135]. It should however be noted that due to framework technical details, the parallelization is not modified accordingly (see Paper IV for further details). The eigenvalue problem is solved by the power iteration method [136].

### 4.1.3 Picard iteration scheme

The implementation of the fixed point iterative scheme discussed in Section 3.3 is presented in terms of a flow chart in Figure 4.3. The scheme alternates between the neutronics and the thermal-hydraulics, applying sub-iterations in each of the modules to increase the overall convergence rate and limit the number of multi-physics iterations.



**Figure 4.3:** Iterative scheme applied for the coupling of the thermal-hydraulics and the neutronics [Paper IV]

The most sensitive part of the simulation is the start of the iterations, and, as seen from Figure 4.3, the algorithm is initiated with an update of the neutronics and the power profile. In general, the thermal-hydraulic equations were slower to converge and found most probable to cause failure of the solver and for this reason the first solution of the temperature equation was delayed to the third multi-physics iteration. However, after a few of these modified iterations, no under-

**Table 4.1:** Geometry specification for the simulated  $15 \times 15$  assembly, with control rod guide tube values in brackets.

Fuel pin radius	0.41 cm
Cladding inner radius	0.43 cm (0.48 cm)
Cladding outer radius	0.49 cm (0.58 cm)
Pitch	1.25 cm
Fuel height	100 cm
Bottom reflector	20 cm
Top reflector	20 cm

**Table 4.2:** Mesh specification for the simulated assembly.

Region	Number of cells
Moderator	6,088,000
Fuel (per pin)	8,000
Cladding (per pin)	4,800
Gap (per pin)	1,600
Neutronics	798,000

relaxation was performed between the multiphysics iterations. An interpretation of this is that the problem is not sufficiently coupled to give stability issues.

## 4.2 Application to a $15 \times 15$ assembly

To exemplify the results produced with the described algorithms, the system presented in Figure 2.11 is simulated. The geometry for the fuel pins is provided in Table 4.1. A summary of the number of computational grids is provided in Table 4.2. As seen from the table, the moderator mesh is significantly finer than the one for the neutronics. Further details in terms of boundary conditions and the domain decomposition are provided in Paper IV, but in overall the thermal conditions are taken to be typical for the conditions in a PWR.

The neutronic calculations were performed for 8 energy groups and with  $S_8$  discretized according to the level symmetric quadrature set. The resulting scalar flux profile of the system is illustrated for the fastest energy group ( $g = 0$ ) and the thermal (lowest energy) group ( $g = 7$ ) in Figure 4.4. In addition to the surface plot, a line plot of the scalar flux at the symmetry line is shown.

Both from the line and the surface plot, artifacts of the so called ray effect can be seen. Such an effect occurs due to the inability of the provided ordinates set to accurately reproduce the angular neutron flux [137, 138] and the solution to this problem is generally to increase the number of directions simulated. The effect was clearly seen in Paper III where a 2D validation of the neutronics was performed against Serpent. Quadratures ranging from  $S_2$  to  $S_{16}$  were compared and the effect was shown to diminish with the increase in the number of ordinates.

DeHart [139] emphasizes that the effect in principle decreases with increasing numerical diffusion (typically due to a coarser grid) and increases with material heterogeneities. In the fine-mesh approach with resolved material regions, both factors mentioned are clearly problematic. In addition, the results from Paper II illustrated the shortcoming of the diffusion solver for the fine-mesh applications as it induced extensive smearing of the neutron flux as compared to the discrete ordinates solver.

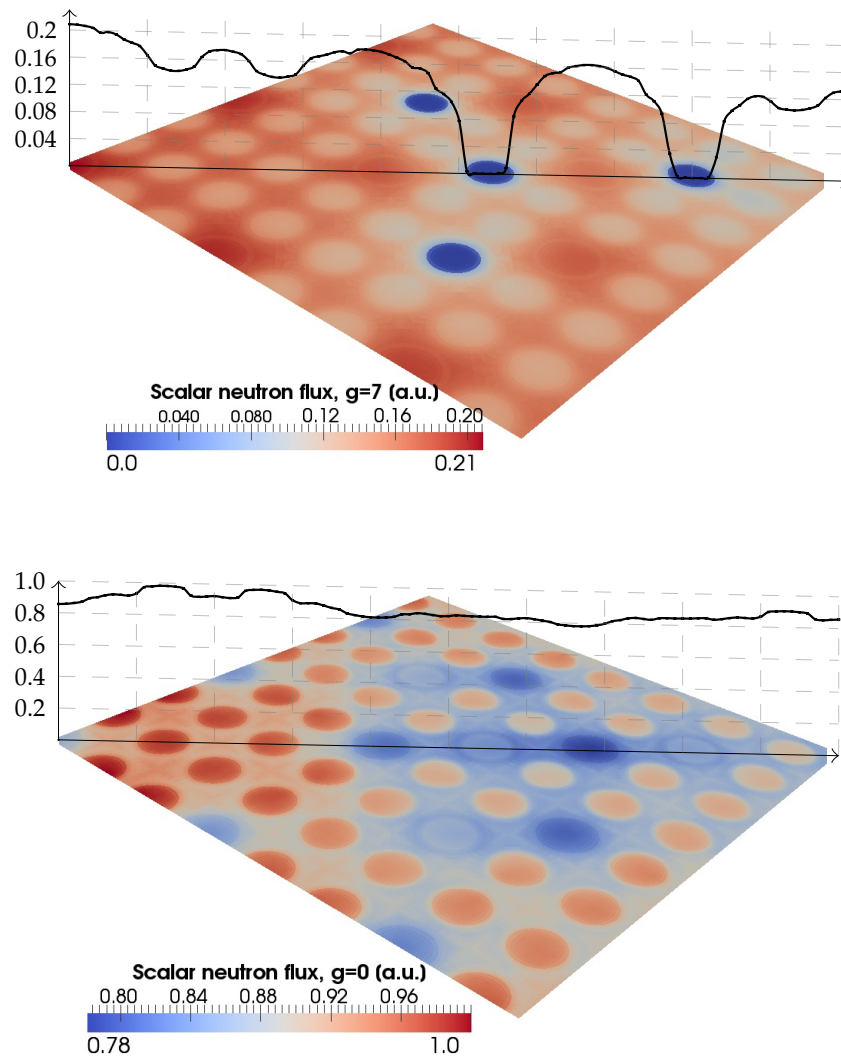
As regards the thermal-hydraulic results, a visualization of the moderator temperature in the assembly is presented in Figure 4.5. As indicated by the plotted horizontal planes, the heterogeneities of the fuel temperature are resolved with the applied mesh resolution. Whereas the total temperature rise for the simulation is a bit more than 10 K for the hot channels, the temperature difference inside one of the sub-channels is less and, accordingly, the density difference for the liquid is small. In turn, such small heterogeneities mean that the influence on the cross-sections is relatively small.

In Paper I, a study was performed on comparing  $k_{\text{eff}}$  computed with a heterogeneous temperature (as in Figure 4.5 but for a  $5 \times 5$  system) against a horizontally averaged profile. It was found that the moderator averaging had no effect on the criticality value. However, the horizontal averaging of the fuel temperature was found to give a significant effect. It should be emphasized that significant simplifications are done during the simulations, including the lack of spacers and the use of the  $k - \epsilon$  model which has clear limitations (as discussed by e.g. [116]). In this context, a time averaging from an unsteady simulation utilizing LES would be of interest to get a second, and potentially more accurate, model for the magnitude of the heterogeneities.

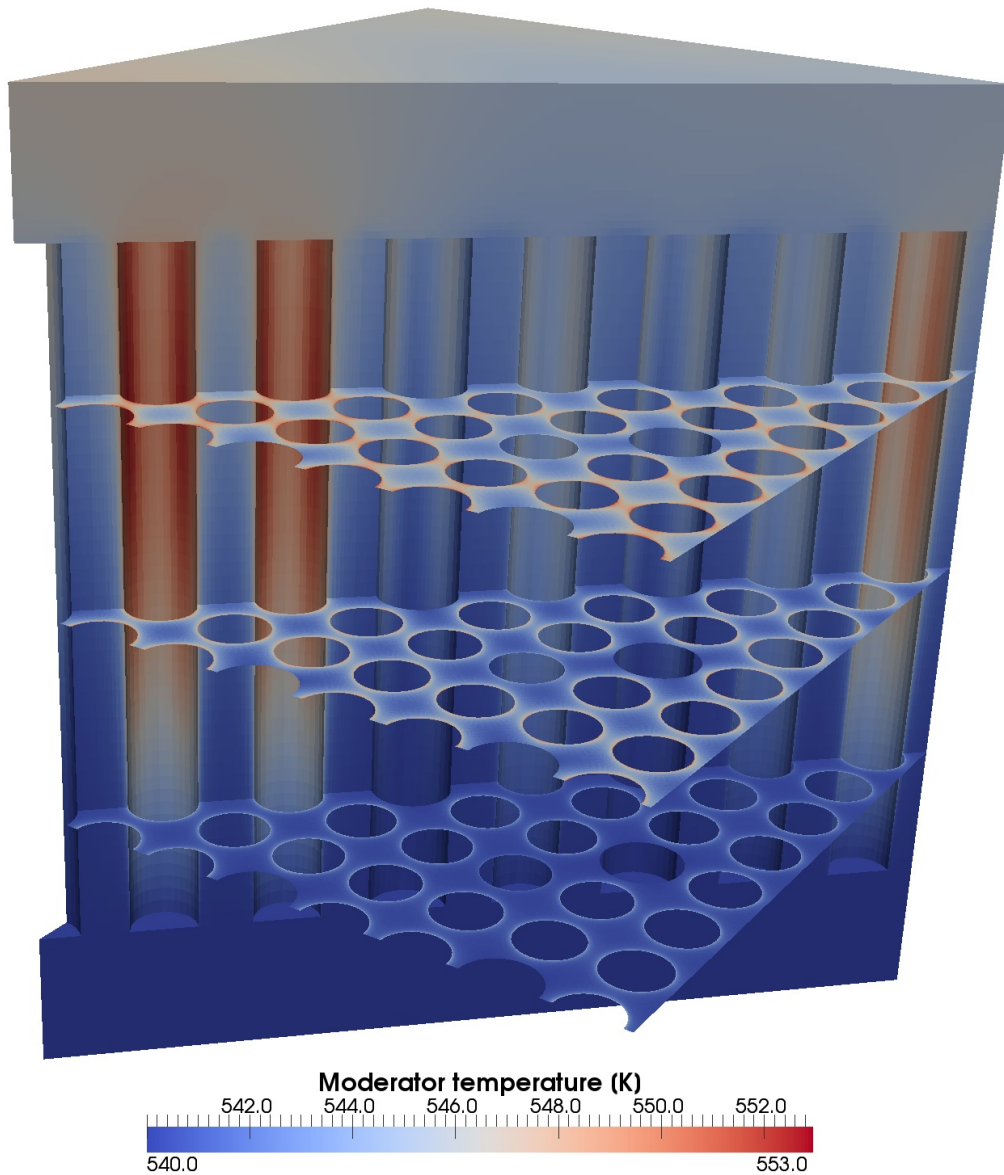
### 4.2.1 Convergence and performance

To evaluate the performance of the schemes for the separate models (Figures 4.1 and 4.2) and the multiphysics scheme (Figure 4.3) the convergence profiles for the multiphysics and the separate module iterations are provided for the first eight outer iterations in Figure 4.6. As described earlier, the first iteration is reserved for neutronics only, followed by a thermal-hydraulic iteration without the temperature equation, and from the third iteration all fields are solved. The iterations following after the 8 first ones require only a single  $S_N$  sweep and the total change in  $k_{\text{eff}}$  from outer iteration 9 to 50 is only 10 pcm (not shown), which suggests that the multiphysics dependencies are more or less fully resolved after the last iteration of Figure 4.6.

As seen from the figure, a limit of 100 sub-iterations is applied for the thermal-hydraulics. Such a limit was found to be increasing the acceleration rate, and in terms of the reasoning on the iterative schemes in Section 3.3, it is reasonable to not directly strive for full convergence for each submodule as the sub-iterations precondition the Picard iterations. In terms of the same section on the coupling, it



**Figure 4.4:** Scalar flux at mid-elevation for the fast group ( $g = 0$ , bottom) and the thermal group ( $g = 7$ , top) for a quarter of a  $15 \times 15$  system as outlined in Figure 2.11 [Paper IV]



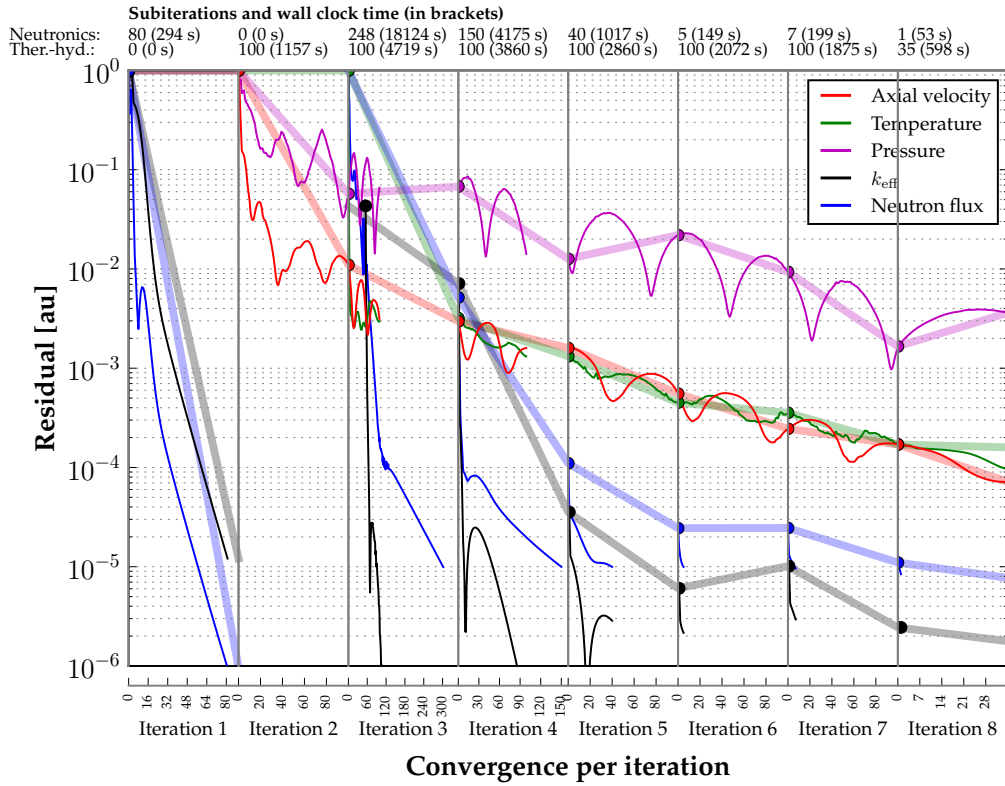
**Figure 4.5:** Moderator temperature at three horizontal planes, with the axial dependence at a diagonal cut in the background [Paper IV].

is interesting to consider the potential benefits of a non-linear solver. As regards the Anderson acceleration, it would be of interest to see the effect on the number of multiphysics iterations. However, as the number of iterations are already relatively few, it is perhaps not to be expected that the acceleration would significantly change the convergence. Instead, as also discussed in Section 3.3, better solvers for the separate fields of physics would likely be more beneficial.

Needless to say, the relative computational time between the neutronics and thermal-hydraulics is strongly dependent on degrees of freedom in the equations, i.e. for neutronics on the number of groups, directions and the grid resolution and for the thermal-hydraulics only on the latter. Nevertheless, it is interesting to com-

pare the simulation effort and we found that for the presented discretization (8 groups,  $S_8$ ) the neutronics and thermal-hydraulics had similar computation time. It should be emphasized that all fields are initiated as flat (i.e. space independent) which is possibly slow but at least giving an honest view of the convergence properties of the system.

As regards the earlier descriptions and discussions on the parallelization implementation (Section 2.1.3), the presented case was run on 64 processors and a steady-state converged solution was computed in a total of 14 wall-clock hours. A full benchmark of the parallelization would require a varied number of cores (strong scaling) or by increasing the problem size (weak scaling).



**Figure 4.6:** Convergence results for the coupled system, with multiphysics iteration convergence as opaque broader lines and the corresponding sub-iterations as thinner lines [Paper IV].





# Transient coupled solver application and analysis

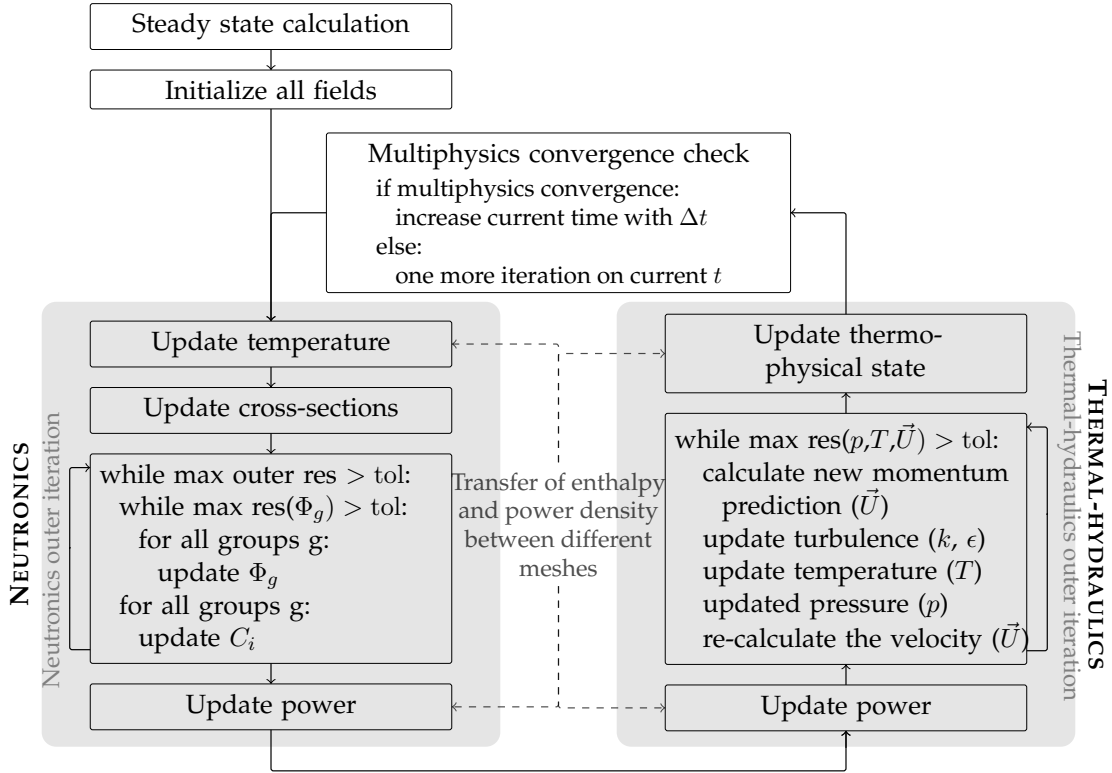
After the steady-state simulations in the previous chapter it follows naturally to continue to transient cases for the same type of systems. For understandable reasons the unsteady models are computationally even more demanding than the steady-state equivalents; instead of converging the multiphysics problem once, all couplings must be resolved in every time step. Nevertheless, the transient cases are potentially of greater interest than the steady simulations as local temporal responses are recovered, and thus an additional contributor to local heterogeneities can be investigated.

This chapter includes implementation details and some of the results from Paper V as well as a description and the results of Paper VIII. In the former, the overall framework is tested for unsteady simulations of a short transient with changing the moderator inlet temperature. The latter proposes a verification method for transient neutronics codes, here applied to a simplified 2D system and shown to give a good agreement with the analytical expressions.

## 5.1 Implementation and framework details

The transient algorithm is implemented with an iterative scheme as outlined in Figure 5.1. In contrast to the steady solver, the transient scheme is purely solving the diffusion equation for the neutronics. Furthermore, the precursor equations are solved by additional iterations. Other schemes are proposed in the literature (see e.g. [140]), where the precursors are implicitly treated directly in the neutron flux equations, which is an area for future improvement. The solution algorithm for the transient solver is again very similar to the steady solver, solved with a segregated approach (PISO).

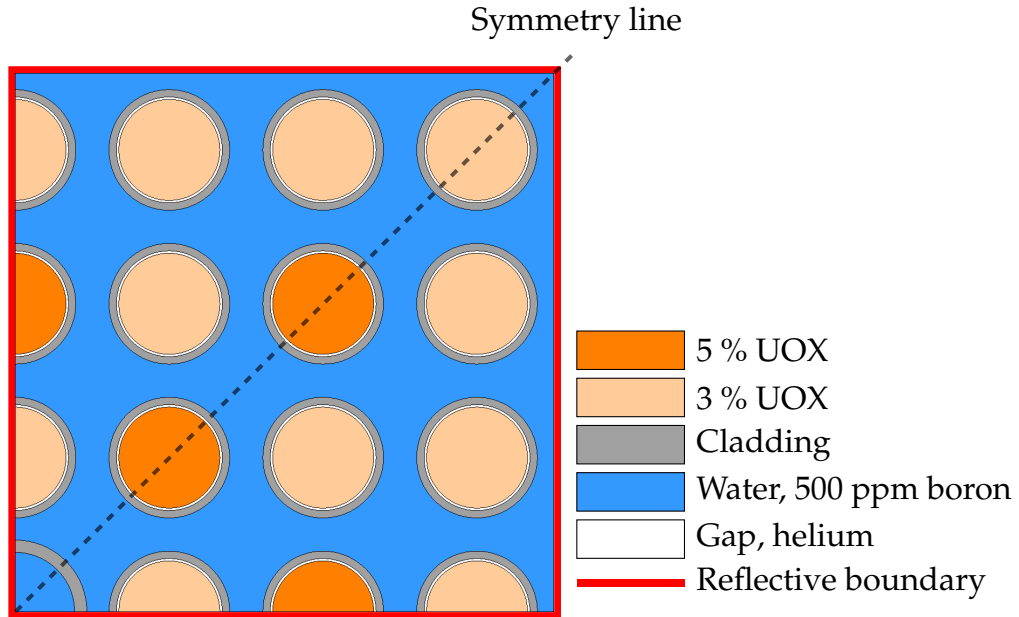
As indicated in Figure 5.1, the transient algorithm is initiated with a solution of the steady multiphysics problem, which was found important for two reasons. First, a good starting guess was found necessary to avoid stability issues with the multiphysics coupling. Second, the system needs to be close to criticality, i.e.  $k_{\text{eff}} = 1$ , at the start of the transient. If the system is far from criticality it either needs to be modified in terms of geometry or material composition, or a renormalization of the fission source needs to be applied. The former remedy was generally found problematic as it requires a re-computation of the cross-sections for a new fuel composition or moderator boron concentration (see Paper V). It should be noted that a desirable (but not implemented) feature would be to include a criticality search in terms of a dynamic boron concentration. For the case of fission source renormalization, the criticality value from the steady simulation is applied as a static (i.e. not changed during the simulation) renormalization. Although such a measure is not physically correct it provides a simple method to test and feature the transient solver.



**Figure 5.1:** Iterative scheme applied to solve neutronics and thermal-hydraulics coupling [Paper V].

## 5.2 Application to a $7 \times 7$ assembly

In Paper V the transient solver was applied to a quarter of a  $7 \times 7$  assembly, again under PWR-like conditions. The horizontal geometry and the material composi-

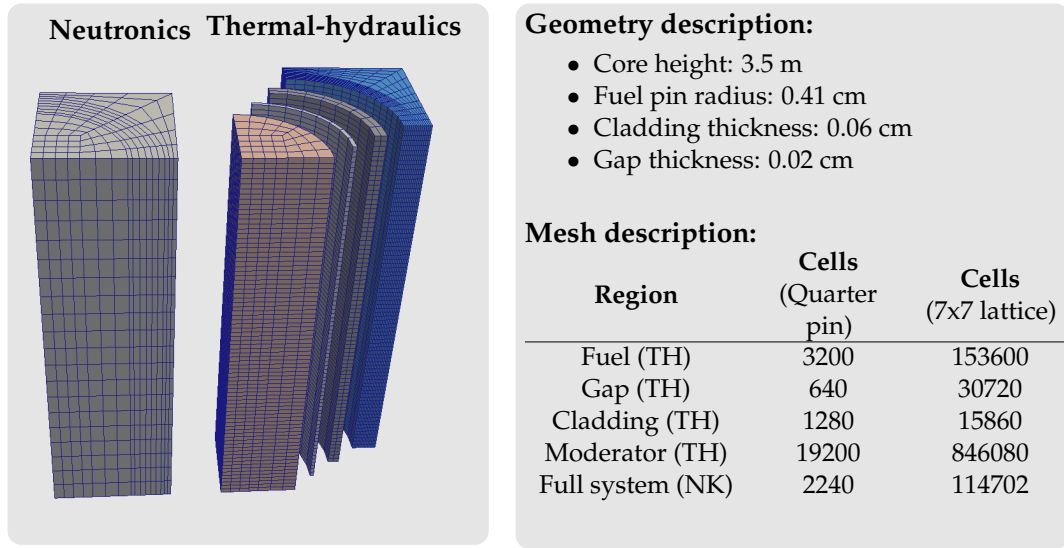


**Figure 5.2:** Horizontal geometry for a quarter of a  $7 \times 7$  system with material regions and the reflective boundary conditions indicated. [Paper V]

tion of the system are outlined in Figure 5.2. Two different enrichments of fuel pins were included and the water contained a boron concentration tested to give a system close to criticality. The overall geometry, the axial mesh resolution and the number of computational cells are provided in Figure 5.3.

The case is run for 10 seconds and between 2 and 3 simulated seconds the moderator inlet temperature is linearly decreased from an initial value of 550 K to 540 K. The uniform temperature decrease at the inlet is propagated through the system and, as an illustration of the response of the system, Figure 5.4 shows the instantaneous moderator temperature (Figure 5.4a) and the fuel power density (Figure 5.4c) after 3 seconds. Similar to the previously presented steady-state results, the subchannel heterogeneities are resolved in the moderator.

In addition, Figure 5.4b shows the temporal development of the relative difference between the maximum and minimum horizontal temperatures in the moderator at mid-elevation of the system. As seen from the figure, the response to the inlet perturbation is not homogeneous, and an increase in the heterogeneity is seen following the temperature decrease. Similarly, Figure 5.4d shows the mid-elevation relative difference in the maximum and minimum horizontal power densities. For the power, the transient leads to an initial decrease in the heterogeneity followed by an increase. Such results would argue that the fine-mesh approach captures the physics not seen when assuming a flat moderator temperature (as e.g. done in a subchannel code). Notably, the magnitude of the difference from the beginning of the transient to the end is relatively small and is arguably of little significance for the presented case.

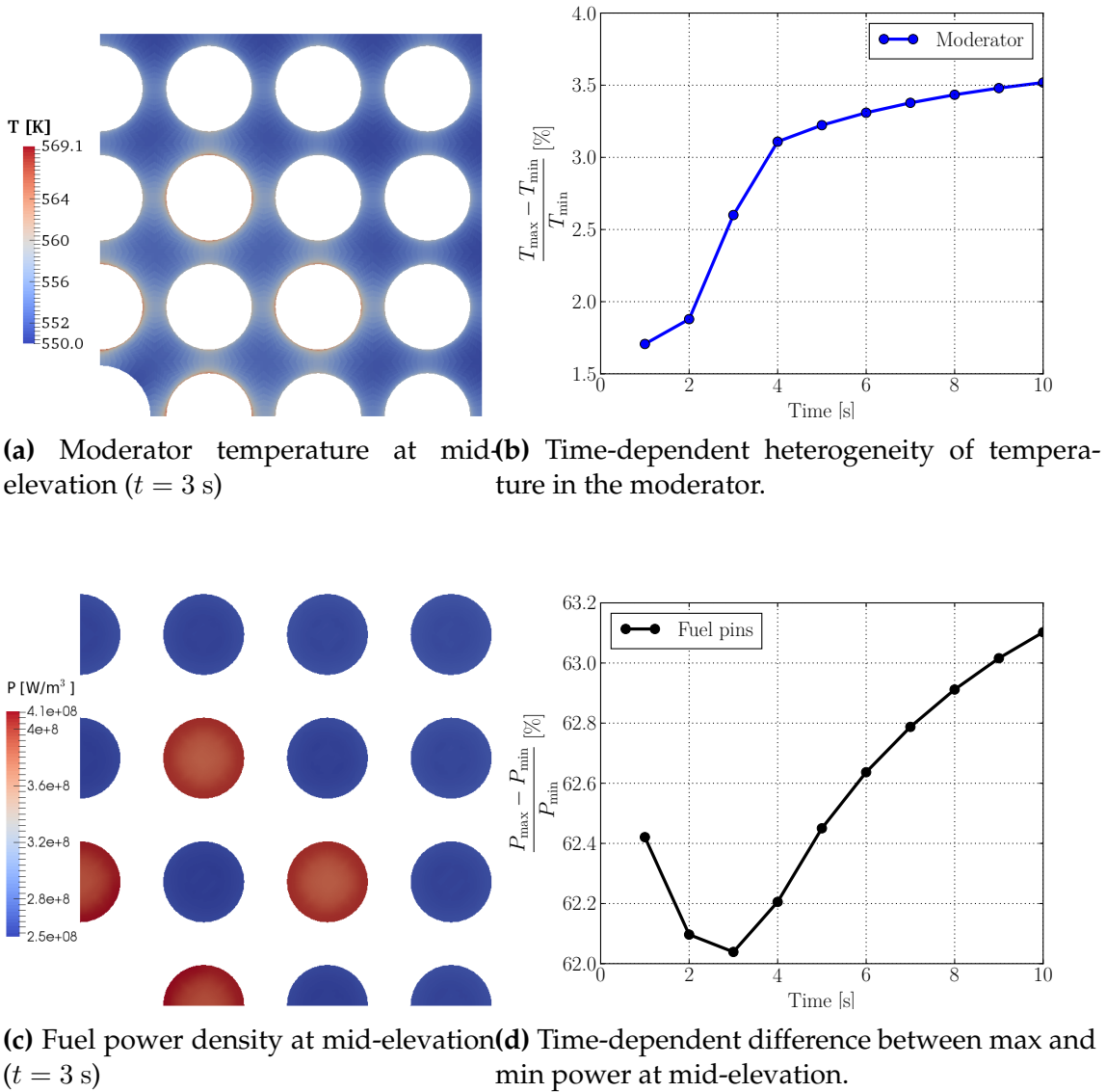


**Figure 5.3:** Description of the geometry and the computational grid for the  $7 \times 7$  system used for transient simulations. Index TH indicates meshes for the thermal-hydraulics, whereas NK indicates the mesh for the neutronics [Paper V].

Considering the computational effort, the case was run on 16 CPUs for a total wall-clock time of 59 hours. In contrast to the case presented in Section 4.2, the time is now primarily spent with the thermal-hydraulics (89%). However, the simulation was here performed with only 4 energy groups and using the diffusion approximation. It is interesting to note that for the majority of the time steps, only a limited number of both the multiphysics iterations and the inner thermal-hydraulics iterations are required. Such an observation suggests that a non-linear formulation would be of limited value for the presented simulation. Moreover, as the applied time step is limited by the Courant number in the CFD simulations an implicit algorithm (e.g. solving all neutronics and thermal-hydraulics equations together in a non-linear fashion) would not necessarily extend the time step.

### 5.3 Time-dependent neutronics verification method

To verify a correct code implementation, in terms of the modeled conservation equations as well as the multiphysics methodology, is typically a difficult task. Except for a direct validation against measurements, codes are not seldom compared and, speciously, verified against each other. The latter is particularly questionable if both codes are based on the same method, i.e. likely to produce the same inaccuracies or similar problems with e.g. discretization. For the fine-mesh multiphysics simulations presented in this thesis, a direct comparison to numerical experiments is difficult. To the knowledge of the author, no equivalent mea-



**Figure 5.4:** Axial slices of moderator temperature and power density with corresponding time development of maximum and minimum horizontal values at mid-elevation [Paper V]

sured data exist. For steady neutronics solvers, verifications against Monte Carlo codes are often performed, and are in some sense an accepted practice in the nuclear field. Such a comparison was done in Paper III for the  $S_N$  solver.

In Paper VIII a method for transient neutronics verification is proposed. The foundation of the method is the application of a stationary perturbation and extraction of the point-kinetic component of the system response. A component of the computed response is verified against an analytical point-kinetic prediction of the same system. The derivation is given in its full detail in Paper VIII, and here only the main characteristics are covered together with an example case.

### 5.3.1 Overview of the methodology

The proposed verification scheme is applicable in both frequency and time space, where the latter approach is exemplified here. The methodology is based on a stationary perturbation applied in one of the fuel regions and in Paper VIII implemented as a sinusoidal variation of the thermal group total cross-section, such that

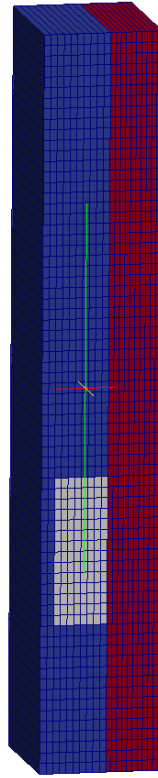
$$\Sigma_{T,2}(\mathbf{r}, t) = \Sigma_{T,2,0}(\mathbf{r}) + A \sin(\omega t), \quad (5.1)$$

where  $\Sigma_{T,2,0}(\mathbf{r})$  is the unperturbed total cross-section of the thermal group and  $A$  and  $\omega$  correspond to the amplitude and frequency of the perturbation, respectively. It should be noted that the derivation presented in Paper VIII was done for a two-group formulation, which is thus followed in the example. In addition to the two-group diffusion equation, the solution to the adjoint problem is required and was consequently implemented and added to the neutronics module.

In order to compute the response of the system, the variation of the power ( $\delta P(t)$ ) and the reactivity ( $\delta \rho(t)$ ) of the system need to be extracted during the simulation. The quantities are computed as volume integrals (i.e. discrete summations over the computational grid) involving the adjoint flux, the cross-sections and the neutron group velocity (for a detailed explanation of the integral expressions refer to Paper VIII). The computed integral values are then fit by a post-processing utility to estimate the phase change as compared to the original perturbation (eq. 5.1) and the amplitude of the responses for  $\delta P(t)$  and  $\delta \rho(t)$ . The fitting function is defined as

$$y(t) = a \sin(\omega t + p) + kt + c, \quad (5.2)$$

where the constant ( $c$ ) and linear variation with time ( $kt$ ) allows for an offset and slow variation of the base line of the fitted functions, which is of importance if the system is not exactly critical at the start of the perturbation.



**Figure 5.5:** Slab system geometry for the example of the point-kinetic verification method. The system is compressed in axial direction, actual length of the system was 50 cm. The blue region indicates the fuel (0.45 cm in width) and the red region indicates the moderator region (0.25 cm in width). The grey region corresponds to the region of the fuel where the time-dependent perturbation was applied. Symmetry boundary conditions are imposed in the horizontal direction. [Paper VIII]

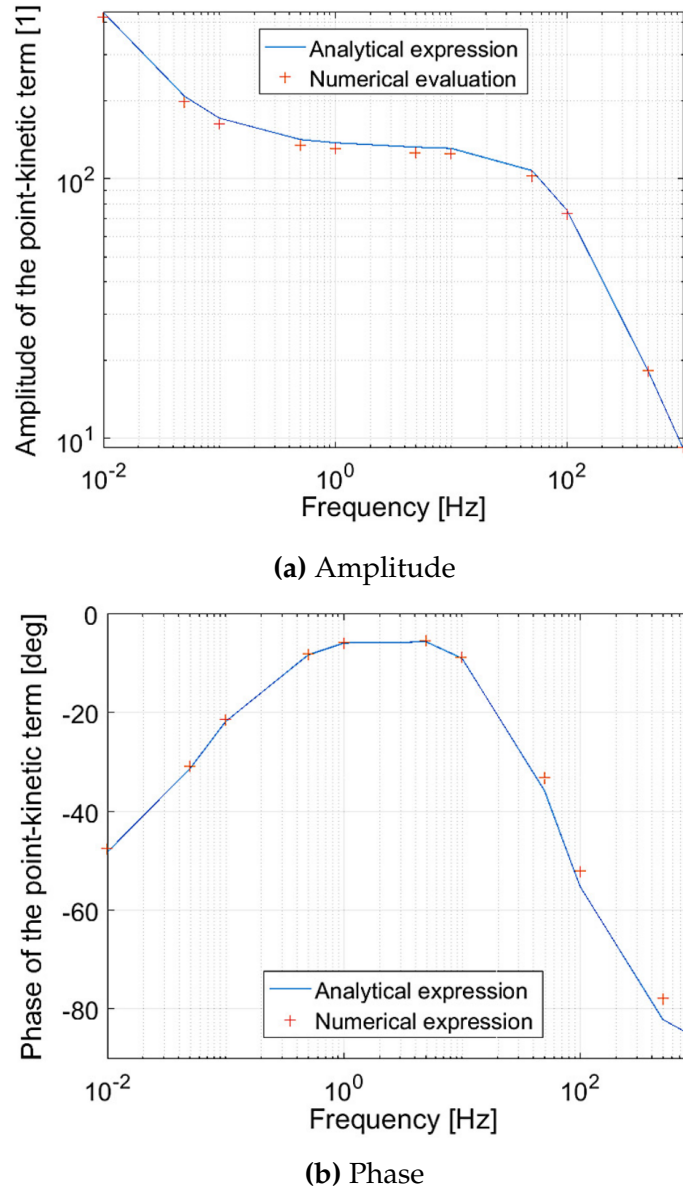
### 5.3.2 Application to a two-region slab system

As an example of the method, a simplistic 2D slab case is presented. The geometry and the computational grid are shown in Figure 5.5. The simulations are initiated from a steady-state multiphysics solution, but performed with a frozen state of the thermal-hydraulics. In addition, a renormalization of the fission source is performed (as discussed in Section 5.1).

To verify the solver, the neutronics solver is run for a range of different frequencies, each providing one point for verification. As a result of the many repeated runs, the verification process is computationally heavy. On the other hand, it was found sufficient to run each simulation for a few periods of the perturbation in eq. (5.1), at least as long as the power is not significantly drifting after the renormalization.

The results of the simulations are provided in Figure 5.6, where the amplitude and frequency of the point-kinetic components of the system response are com-

pared to the analytical solution of the point-kinetic equation. As seen from the figure, the results of the FVM code match both the amplitude and phase well, with a maximum deviation in the amplitude of less than 5%. Such a result is important not only for the presented solver but also as an example of a method with great potential to other 3D transient neutronics codes, thus filling an earlier vacuum in terms of lacking verification methods.



**Figure 5.6:** Frequency dependence of the amplitude (top) and the phase (bottom) of the point-kinetic zero-power transfer function for the slab presented in Figure 5.5 [Paper VIII]



# On two-phase flow in LWRs

Up to this point in the thesis the coolant of the reactor was considered to be single-phase liquid water. Such a flow is representative for nominal conditions in a PWR, but not at all for describing the flow in a BWR. In the latter, the water enters the core in a subcooled liquid phase but boils and over the height of the channel it traverses a number of flow regimes. Whereas single-phase flow is relatively well understood and well reproduced with simulations, two-phase flow is still, after more than half a century of studies, a very challenging topic.

In this chapter, the two-phase flow problem of liquid and vapor water in the BWR is briefly introduced (Section 6.1), with primary focus on different strategies for simulation and modeling (Section 6.2). Due to the difficulties of formulating a single universal model covering all regimes and scales in a computationally feasible manner, the problem inevitably needs to be narrowed for the scope of this thesis. In the current work two aspects are of primary focus, namely simulation of subcooled boiling flows (Section 6.3) and dynamic characteristics of the two-fluid model (Section 6.4).

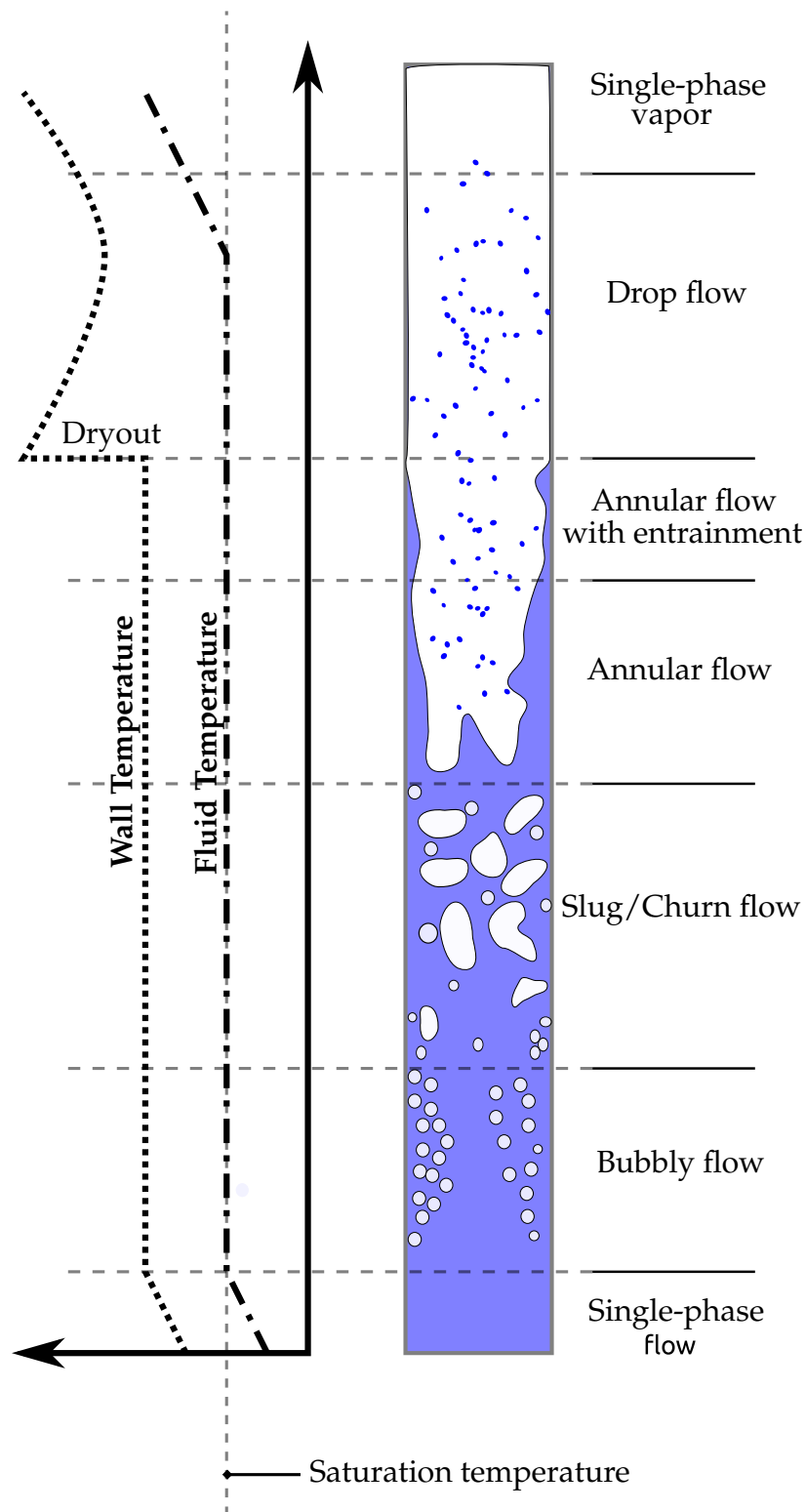
In relation to the previous two chapters on the application of the multiphysics tool, the current chapter is in a sense more generic. The simulations are performed on simplistic geometries and in parts reduced in terms of model complexity. To a large extent this is done to better illustrate the proposed ideas, and limit the distraction of the huge number of force models, flow regime correlations, etc. Furthermore, the purpose of the seemingly reduced models is to clarify some of the underlying complexities in the models often foreseen in routine application of multiphase CFD.

## 6.1 Perspectives on the simulation challenges

The methods to simulate multiphase flows are in many sense as multifaceted as the configurations of the flow systems themselves. To model a particulate flow of gravel requires completely different strategies as compared to a slug flow in a channel of the BWR. Needless to say, the difficulties are accordingly distinct for each problem and undoubtedly multiphase flow systems constitute some of the most outstanding challenges in the field of fluid mechanics.

The BWR core is an important example of the intricacies of two-phase flow in industrial devices. To extend on the complexities and the motivations for CFD simulations of BWR subchannels, Figure 6.1 is used as an illustration. The figure shows a heated channel with flow regimes ranging from bubbly flow to the extreme of single-phase vapor. Some of the reasons for complications include the phase changes due to boiling and condensation, the different characteristics of the flow regimes and the issues with overlapping scales of the phase heterogeneities and the geometry in the narrow channels. In some more detail:

- *Phase change* – The energy released from fission in the fuel heats the water entering the core to the saturation temperature and induces boiling at the cladding surface. The liquid phase reaches saturation conditions close to the wall after only a short axial distance (see the sketch in Figure 6.1), whereafter the water boils at the wall. In addition, during the subcooled phase the bubbles are transported from the wall to the bulk of the flow and they condense, which is one of the phenomena targeted in Section 6.3.
- *Flow regimes* – As the vapor phase continues to increase, the flow regime in the channel changes. The initial bubbly phase of vapor is transformed into larger regions of void, depicted in Figure 6.1. If the flow is further heated, the regime might even become the reverse of the initial bubbly flow, where instead the liquid phase is dispersed as droplets in the vapor bulk. It should be noted that the latter conditions, starting from the point of dryout, must by all means be avoided as there is a risk to melt the fuel due to the severe decrease in the heat transfer from the fuel.
- *Separation of scales* – In contrast to the single-phase flow regime, separation of scales is not trivially fulfilled in the case of multiphase flow. In short, the lack of separation of scales is a result of that the size of the void regions approaches the size of the computational grid. As an illustrative example consider the slug regime in Figure 6.1. It is readily imagined that in order to resolve the velocity profile of the depicted channel, the computational cells would need to be on the size of (or rather even smaller than) the void structures. Such an issue has been a long standing challenge of multiphase flow [142], and due to the interfering scales it becomes difficult to formulate a space averaged model to resolve fine scales of the simulations (such as the flow inside the BWR subchannel) while still fulfilling the separation of scales [143].



**Figure 6.1:** Overview of forced convection boiling phases in a channel (in part redrawn from [141]).

As indicated in the motivation for the fine-mesh simulations (Section 1.3.1), numerical predictions of CHF are both interesting and important challenges in CFD simulations of BWRs and a major driver for the development of new models [144]. To accurately model CHF, multiple scales are relevant, ranging from the micro scales of the growth of a bubble on the surface of the cladding to the transport of bubbles (or larger chunks of void) in the subchannels, and to the scales of the full fuel assembly. For the first, the growth of the bubble and the departure from the wall, very detailed descriptions of the flow are generally required (see e.g. [145]) and this type of simulations fall outside the scope of the current work. For the latter two scales (bubble departure and the entire subchannel), the focus is primarily on the flow of two phases and the potential phase change inside the fluid which are the topics of the next section, introducing some modeling approaches of the flow problem.

## 6.2 Models for two-phase liquid and gas flows

In contrast to the subchannel approach (Section 1.2.2), which to a large extent is driven by empirical, macroscopic relations and tuned models, the fine-mesh or CFD approach focuses on a physics-based modeling of the flow. In coarse terms, the methods could be divided into two groups: interface tracking methods and averaging methods. The former is sometimes denoted DNS-like methods and the latter is often deployed as the two-fluid model.

### 6.2.1 DNS-like methods

The concept of DNS in multiphase flow is not as straightforward as in the single-phase case. In the latter, the Navier-Stokes equations are resolved on every scale down to the smallest fluctuations in the flow [116], with small risk of adventuring the separation of scales. For multiphase flow, DNS is occasionally referred to as simulations in which the interfaces of the multicomponent flow are resolved. However, the exact nomenclature has for sure been debated (see for example the note by Yadigaroglu [146]), and it should be noted that the increase in computational resources has enabled bubbly flow DNS simulations in the single-phase sense, i.e. with full resolution of each of the phases [147].

Even though the flow fields can be fully resolved with the Navier-Stokes equations without modifications or models, there are still fundamental challenges with phenomena such as bubble coalescence and breakage related to the effects of surface tension. For the latter, a certain amount of modeling is still required (see e.g. [147]). Furthermore, the systems that can be resolved in a DNS manner are still small, extremely computationally demanding and primarily focused on the bubbly flow regime. Consequently, a simulation of a full BWR channel with anywhere close to realistic void fractions is still out of reach. Nevertheless, DNS

like simulations are of interest to formulate models for macroscopic correlations, or to perform multiscale hierarchal simulations (such as reported for gas-solid flows [148]).

Despite that the BWR subchannel problem does not easily lend itself to full DNS calculations, interface tracking methods (without all length scales resolved) are still of interest and relevance. A range of different methods have been proposed, whereof the most notable include the Volume of Fluid method (VOF) [149, 150], the Level Set method (LS) [151, 152] and front tracking [153]. Interesting examples include single bubble condensation [154, 155], which is potentially valuable to formulate correlations for a coarser (two-fluid) model. The DNS simulations alike, the computational effort required also by coarse applications of interface tracking methods precludes their use for full subchannel simulations. For larger scale CFD simulations an averaging method is instead required.

## 6.2.2 The two-fluid model

The two-fluid model describes the two phases in an Eulerian-Eulerian frame of reference [143, 156] and instead of tracking the interface, the phases are treated as interpenetrating continua. The presence of the phases is described in terms of a volume fraction, for liquid-vapor systems typically denoted void fraction. Due to the fact that the interface is not directly tracked, the computational burden of the method is much smaller than in the case of DNS-like simulations. On the other hand, the averaging comes with a price. In comparison to the DNS-like methods described in the previous section, a larger degree of modeling is required, in particular for the interphase mass and momentum exchanges.

The governing equations of the two-fluid are here briefly outlined for the purpose of the applications in Sections 6.3 and 6.4, but without a detailed derivation (for a detailed discussion on the procedure see e.g [6]). The model relies on one or more averages on the Navier–Stokes equations for each of the phases. Such an averaging can be performed as volume averages (see e.g. [6, 157]), ensemble average (see e.g. [158, 159]) and/or in terms of time-averages. The result is a set of mass, momentum and energy conservation equations for each of the phases. In the current work the mass conservation is given as [157]

$$\frac{\partial \alpha_i \rho_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i) = 0, \quad (6.1)$$

where  $\rho_i$  is the density,  $\alpha_i$  the phase fraction and  $\mathbf{U}_i$  is the velocity, and the momentum equations as

$$\frac{\partial \alpha_i \rho_i \mathbf{U}_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \mathbf{U}_i \mathbf{U}_i) = -\nabla \cdot (\alpha_i (\bar{\bar{\tau}}_i + \bar{\bar{\tau}}_i^t)) - \alpha_i \nabla(P) + \alpha_i \rho_i \mathbf{g} + \mathbf{M}_i, \quad (6.2)$$

again written for each of the phases and where  $\bar{\bar{\tau}}_i$  and  $\bar{\bar{\tau}}_i^t$  are the viscous and turbulent stress tensors, respectively,  $P$  is the pressure and  $\mathbf{M}_i$  represents the interfacial momentum transfer. Furthermore, conservation equations for energy are

required for diabatic simulations, and in the case of boiling or condensation additional terms appear in all the equations. The latter is the case in Paper VI, whereas only eq. (6.1) and (6.2) are solved in the stability and dynamics investigations in Paper III and VII. The set of equations (6.1) and (6.2) is commonly solved in a similar manner as the PISO algorithm for single-phase flow. The implementation utilized in the thesis closely follows the derivations by Rusche [160] and Weller [159].

As already mentioned the two-fluid model suffers from a number of shortcomings. The following (non-exclusive) list of issues and limitations is notable:

- *Lack of stability* – It has previously been reported that the two-fluid model in a pure form (i.e. without additional artificial or physical model-based viscosity) suffers from lack of hyperbolicity, which might lead to instable behavior of the simulation [161, 162]. Such instabilities are the topic of Paper III and Paper VII discussed further in Section 6.4.
- *Lack of size distributions* – As a result of the averaging, the specific information of the phase interface is lost. Consequently, for the example of a bubbly flow the bubble size distribution is unknown and as a result all size-dependent correlations (e.g. momentum exchange terms and the condensation rate) cannot be accurately applied. A potential remedy for this is to regain size distribution information from a PBE, which is the topic of Paper VI.
- *Lack of separation of scales* – As the two-fluid model equations are commonly derived under the assumption of small variations of all fields (at least for volume averaging) only low void fractions or small bubbles could theoretically be treated. This is commonly abused as the computer codes often run also for higher void fractions. Arguably, carrying out ensemble average is advantageous since separation of scales is not an immediate issue for that procedure. However, in practical cases the interphase exchange terms are typically implemented in a volumetric sense [163], thus again a prey for the mentioned issues. Interestingly there are some (theoretical) examples of derivations not requiring separation of scales/small local gradients [164, 165], which are of future interest.

From a historical perspective the two-fluid model, in a lower dimensional formulation, has been the standard choice for the system codes to describe the multi-phase flow. Also in so-called mechanistic CFD modeling, i.e. 3D simulations with (more or less) well-founded physics-based models (see e.g. [166]), the two-fluid framework has been and still prevails as the dominant methodology. By large the two-fluid model should perhaps be seen as the workhorse of the two phase simulations, and although in many senses imperfect, it enabled interesting 3D calculations already 20 years ago [167].

In terms of the first part of this thesis, the fine-mesh multiphysics framework, the two-fluid solver is a good candidate for coupling to the neutronics. In par-

ticular, the possibility to simulate a complete subchannel is relevant. A DNS-like simulation of only a few bubbles is arguably too small to be of relevance for the coupling.

## 6.3 Population balance for subcooled bubbly flow

As described in the previous section, the averaging of the Navier–Stokes equations for the two–fluid model results in that the microscopic details of the flow are lost. For a bubbly flow, as depicted in the region close to the inlet of the channel in Figure 6.1, the loss of interface information between the vapor and the liquid phase results in an unknown distribution of bubbles. In practical terms, this implies that the two-fluid simulations are performed for a single size and shape of bubbles, or potentially estimated from an average interfacial area concentration [168]. A remedy for the information lost in the averaging is to apply a PBE to retrieve additional knowledge of the state of the dispersed phase. For gas-liquid flows it is common to track the bubble size or volume [169–171], but in principle other parameters such as bubble shape or velocity could also be described with the PBE.

In particular, for simulations of the onset of boiling and the transport of bubbles in a subcooled liquid the size distribution has been argued to be of importance [172]. Such a distribution is applicable both in the description of the condensation in the liquid and the aggregation and breakage of the bubbles. Thus, a subcooled bubbly flow is an interesting candidate for PBE simulations. In the nuclear community, the primary choice for subcooled boiling simulations with PBE has been the Multiple-size-group method (MUSIG), which represents the size distribution with a fixed set of, a priori determined, sizes. MUSIG has previously been successfully applied to subchannel simulations [172–174].

An alternative approach to solve the PBE is given by the Direct Quadrature Method of Moments (DQMOM), which relies on dynamic sizes allowed to vary dynamically throughout a simulated domain [175]. An advantage with DQMOM over MUSIG is that the use of non-fixed sizes allows to describe the distribution with the same accuracy for fewer degrees of freedom [170, 176]. DQMOM has previously been applied for adiabatic cases in the field of nuclear engineering [171] and in other fields for evaporation simulations [177]. In Paper VI, a formulation for the DQMOM of condensation of bubbles is proposed, implemented and compared to MUSIG. To give relevance to the example provided in Section 6.3.2, a brief overview of the methodology is given in the following section.

### 6.3.1 PBE formulation

The PBE for a bubble size distribution is written as [178]

$$\frac{\partial f(\xi, \mathbf{r}, t)}{\partial t} + \frac{\partial}{\partial \xi} \left( \frac{\partial \xi(\mathbf{r}, t)}{\partial t} f(\xi, \mathbf{r}, t) \right) + \nabla \cdot (\mathbf{U}(\mathbf{r}, t) f(\xi, \mathbf{r}, t)) = S(\xi, \mathbf{r}, t), \quad (6.3)$$

where  $f$  is average number density,  $\xi$  is the length scale (diameter) of the bubbles,  $\nabla$  refers to the convection in space and  $S$  is a source term which appears due to condensation and aggregation of bubbles. The velocity in eq. (6.3) is computed from the momentum equation for the vapor phase and is in the current methodology independent of the bubble size. To simulate condensation, the second term on the left hand side is written in terms of the condensation as

$$\frac{\partial \xi(\mathbf{r}, t)}{\partial t} = C(\xi, \mathbf{r}, t). \quad (6.4)$$

It is particularly important that the condensation model is allowed to have a size-dependence as empirical models typically introduce the size in the correlations [179]. As briefly mentioned, a two-fluid solver not complemented by the PBE is limited to a static size of bubbles.

### DQMOM

In DQMOM, the average number density is discretized in terms of  $N$  abscissas ( $\xi_i$ ) and weights ( $w_i$ ) such that

$$f(\xi; \mathbf{x}, t) \approx \sum_{i=1}^N w_i(\mathbf{x}, t) \delta(\xi - \xi_i(\mathbf{x}, t)). \quad (6.5)$$

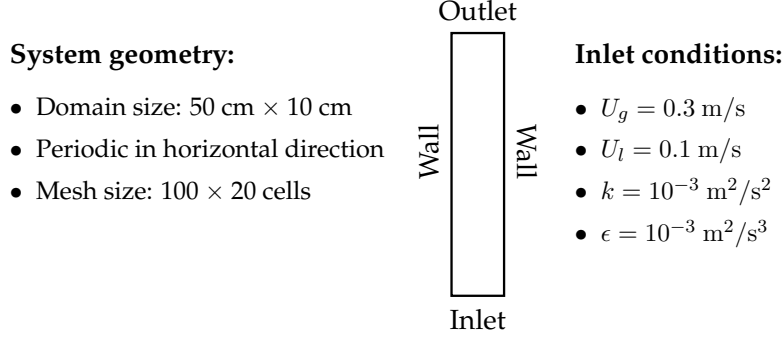
To close the set of equations for the weights and abscissas, a moment transform is applied to eq. (6.3). The transform results in coupled transport equations for the abscissas and the weights based on the  $2N$  first moments of the PBE. An advantage of DQMOM over some similar, moment based, methods is that the equations are relatively easy to implement, only requiring a discretized time derivative and convective term. The source terms of the transport equations are computed from a cell-wise linear system of size  $2N \times 2N$ , and to resolve the couplings between the equations an iterative scheme is applied. For further computational details the interested reader is referred to Paper VI.

### MUSIG

As a reference for the proposed DQMOM formulation for subcooled boiling, a MUSIG model was implemented. The method is entirely based on models from the literature, where the PBE is often written in terms of bubble mass. Accordingly, eq. (6.3) is formulated as a set of vapor fraction equations

$$\frac{\partial \alpha_{g,j} \rho_g}{\partial t} + \nabla \cdot (\alpha_{g,j} \rho_g \mathbf{U}_g) = S_j, \quad (6.6)$$





**Figure 6.2:** Geometry and boundary conditions for the rectangular, horizontally confined system applied in Case 3 of Paper VI.

where  $j$  is the number of the class,  $S$  is again a source term and  $U_g$  is common to all bubble sizes. For MUSIG the condensation is implemented as a source term coupling the equation for the current  $j$  to the sizes below and above ( $j - 1$  and  $j + 1$ ).

### 6.3.2 Application to channel flow with condensation

To exemplify the DQMOM formulation, the case presented in Figure 6.2 is studied. The simulation domain is a 2D channel with no-slip conditions for the liquid at the horizontal walls. A small superficial velocity is applied for the bubbles and the inlet liquid temperature is subcooled by 1 K as compared to the saturation temperature. Furthermore, all thermophysical properties are computed by interpolation from tables handling both the pressure and temperature dependence. Finally, the inlet bubble size distribution is computed according to a normal size distribution with an average size of 7 mm bubbles.

The PBE methods (DQMOM and MUSIG) are both coupled to the two-fluid model with all terms handling the phase change due to condensation included. As the convection of the bubbles is directly simulated by DQMOM/MUSIG, the continuity equation is not explicitly solved in the two-fluid solver. The model includes drag, virtual mass, lift and turbulent dissipation interphase forces, where the  $k - \epsilon$  model is used to solve the turbulence in the liquid phase. For the mentioned forces, the size distribution is used to compute the total momentum force as a sum of the contributions from each abscissa or class for DQMOM and MUSIG, respectively. In addition, the condensation model has a size dependence as previously discussed.

Example results of the simulation are presented in Figure 6.3. The top plot (Figure 6.3a) shows the development of the void fraction over the simulated channel. As seen, the initial void fraction quickly decreases as the vapor bubbles condense and shrink. The latter effect is detailed in the bottom plot (Figure 6.3b)

where the average size of the bubbles in the system is computed as

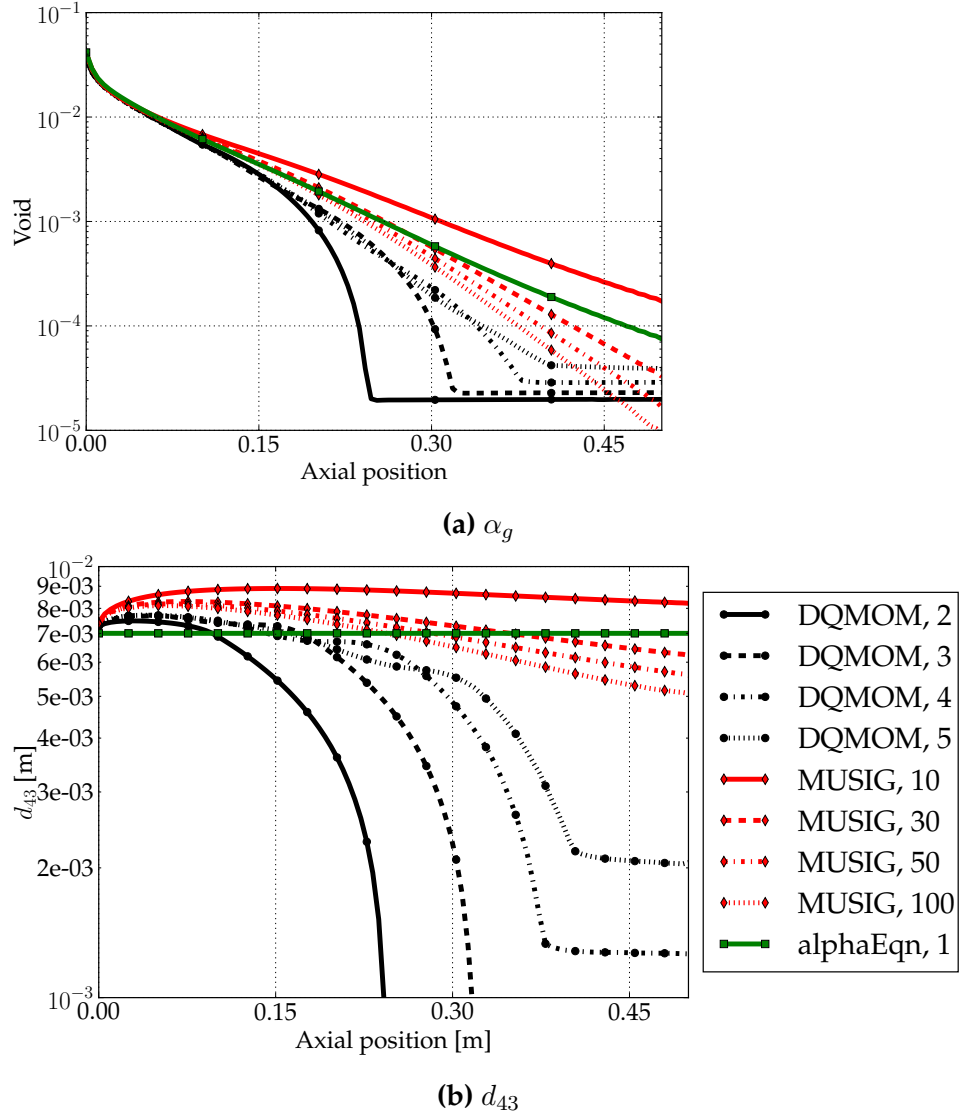
$$d_{43} = \frac{\sum_{i=1}^N \xi_i^4 w_i}{\sum_{i=1}^N \xi_i^3 w_i}. \quad (6.7)$$

As seen from the figure, the shrinkage of the bubbles accelerates over the channel, particularly visible for DQMOM, and the effect is explained by an inverse proportionality of the condensation rate with the bubble size. Due to the fixed sizes in the MUSIG method, a much smaller range of bubble sizes is covered and this was in Paper VI shown to be a severe limitation, further emphasized by other examples in the mentioned paper. It should be noted that both for the void fraction results and for the average size, DQMOM reach minima at around half the distance of the channel. For intricate reasons of the formulation, a threshold size of the abscissas is needed to avoid numerical issues. However, as seen from the figures, such a remedy occurs only at void fractions which are too low to be of physical significance.

In addition to the example above, different studies of bubbles inserted at the walls were conducted and a formulation useful for wall boiling models with DQMOM was proposed. Wall boiling is a particularly challenging topic as the insertion of bubbles takes place locally at the wall and in the discretized domain only in the first layer of cells.

Furthermore, under even more simplistic conditions, DQMOM was shown to converge for much fewer abscissas than the number of classes needed in MUSIG. For a case without the coupling to the two-fluid solver and with an initial mono-size distribution, MUSIG is shown to require more than 100 classes to capture the size change predicted with only a few abscissas. The difference in number of required sizes also results in a significant difference in the computational time which was, advantageously, shorter for DQMOM. However, for the examples coupled to the two-fluid solver, the differences in the computational effort are smaller. In practical terms MUSIG seems to have advantages in the simpler implementation and generally more stable characteristics, whereas the primary advantage of the DQMOM is the feature of dynamic abscissas which can cover very different ranges of bubble sizes in different parts of the domain.

As clear from the system description of the example case the intention in Paper VIII is neither to directly mimic the geometry nor do reconstruct the exact conditions in a subchannel in a BWR assembly. Instead, the purpose is to propose and evaluate a potential candidate for PBE simulations, which is also closely related to the second objective outlined in Section 1.4.2. In addition, the evaluation forced additional studies of the coupling between the two-fluid model and the PBE methods which are of future value for simulations dedicated to the actual BWR problem.



**Figure 6.3:** Vapor fraction (top) and average bubble size (bottom) along the axial centerline compared between MUSIG and DQMOM and for a void fraction equation (labeled alphaEqn) with a single static class. [Paper VI]

## 6.4 On the dynamics of the two-fluid formulation

As suggested in the introduction to the two-fluid model, the formulation is known to be prone to instabilities. In particular from the nuclear perspective, the 1D version of the conservation equations with no viscosity has been reported to have issues with such instabilities [161]. From a mathematical point of view, it is well known that the degree of hyperbolicity in the equations affects the stability and, for the same 1D formulation, mathematical regularization of the problem has been proposed as a remedy to achieve hyperbolicity [162]. Another approach often seen in the literature is numerical regularization, basically achieved by inducing numerical diffusion, either through the use of a coarse spatial discretization or in many papers, implicitly, by a diffusive turbulence model.

The potential instabilities are not only interesting for the 1D formulation, but also for the fine-mesh 3D approach in the current thesis. In particular, the influence of such instabilities on the dynamics of the two-fluid model is of interest. Although the phases are only predicted in an average sense, the dynamics of the fields are influential for mass and heat transfer applications. Additionally, it is of importance to understand any potential discrepancy between instabilities induced only by numerical issues and the heterogeneities actually predicted with experiments (such as in [180]). Numerical experiments in terms of 3D simulations with the two-fluid model have previously been studied from a stability perspective for gas-solid flows. It has been shown that such flows exhibit the so-called meso-scale instabilities, i.e. regions with fluctuations smaller than the physical domain but larger than the characteristic size of the particles [181].

From the literature on two-fluid simulations for gas-liquid problems it is clear that there are still a large number of open questions, and the formulations of e.g. momentum exchange mechanisms are basically as many as the authors. In addition, the treatment of turbulence for such systems is an area with a lot of proposed methods reflected in additional terms for the momentum conservation equations. A comprehensive discussion on such terms and on the lack of consensus is provided in Paper VIII.

In the work done for the thesis, the studies of the dynamics of the adiabatic two-fluid model are deliberately based on a much simplified model. The approaches reported in Papers III and VII are to prune the studied equations of any additional (unnecessary) terms and apply the model to simplistic cases. In Paper III, the two-fluid model according to eqs. (6.1) and (6.2) is simulated including only the drag term for the momentum exchange between the phases. For Paper VII the drag-only simulations are extended and compared to simulations also including the virtual mass force. In specific, the performed simulations are an attempt to understand the possibilities of resolving the dynamics of the gas-liquid flow with the two-fluid model. Additionally, it has been the goal to formulate, with physical or numerical arguments, criteria to discern numerical issues from instabilities of physical origin.

As regards the presented results, it should be noted that the two included papers are based on different CFD solvers (ANSYS Fluent and OpenFOAM®). Furthermore, variations of the implementation of the continuity equations was studied elsewhere [182]. In Paper III a 2D system was studied and the results from the paper are omitted here. Instead, the brief discussion in the next section is based on the results from the 3D system in Paper VIII.

### 6.4.1 Application to adiabatic cases

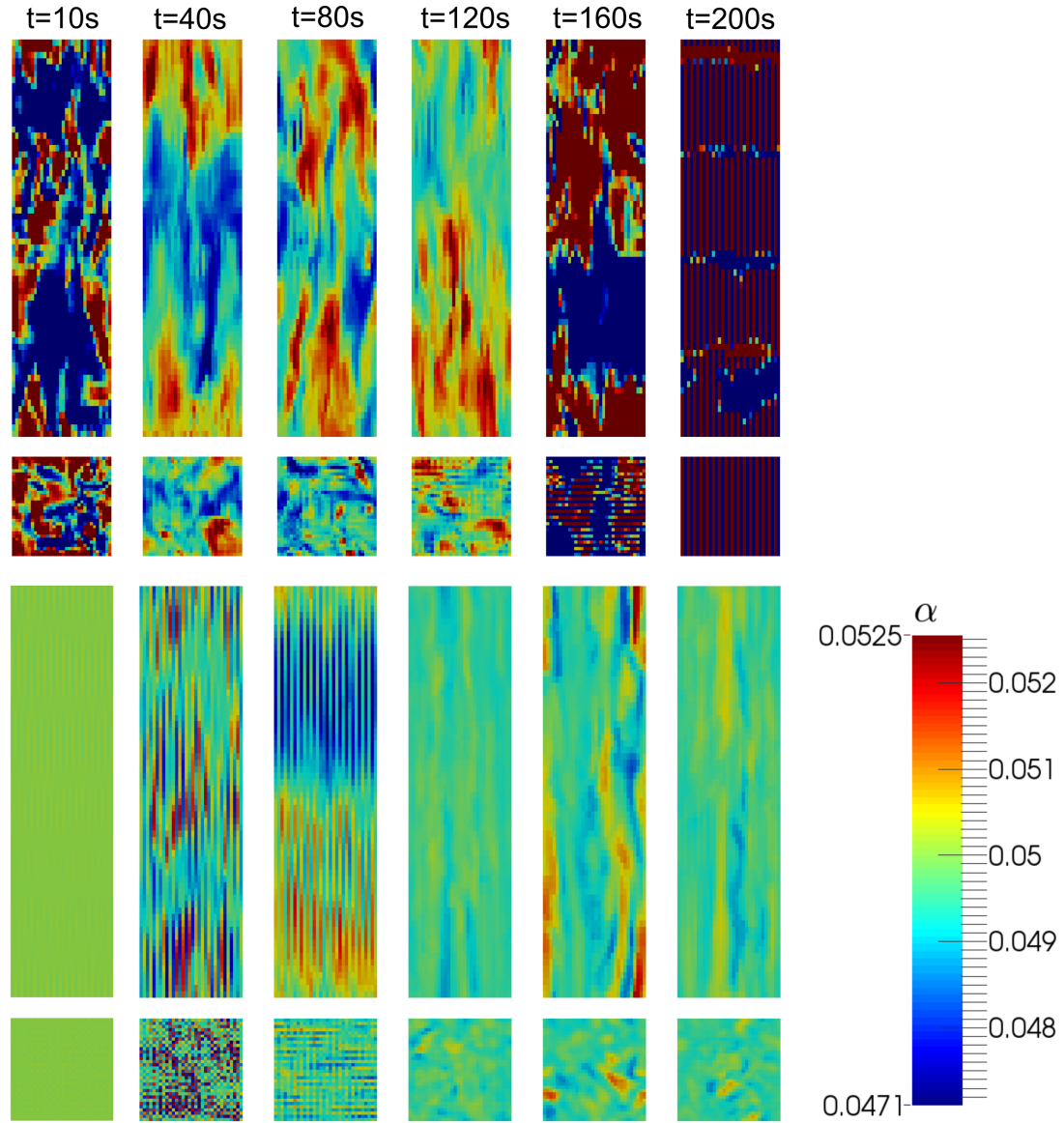
The cases studied are simple in the sense that only adiabatic conditions are considered and the simulated domain is periodic in all directions. The effect of gravity is included and to outweigh the combined mass of the gas and liquid phases a jump condition is applied for the pressure. The initial fields and the thermophysical properties are presented in Table 6.1. All fields are initiated with spatially uniform values, i.e. no initial perturbations are applied to induce the instabilities.

**Table 6.1:** Thermophysical parameters and initial conditions as applied in Paper VII.

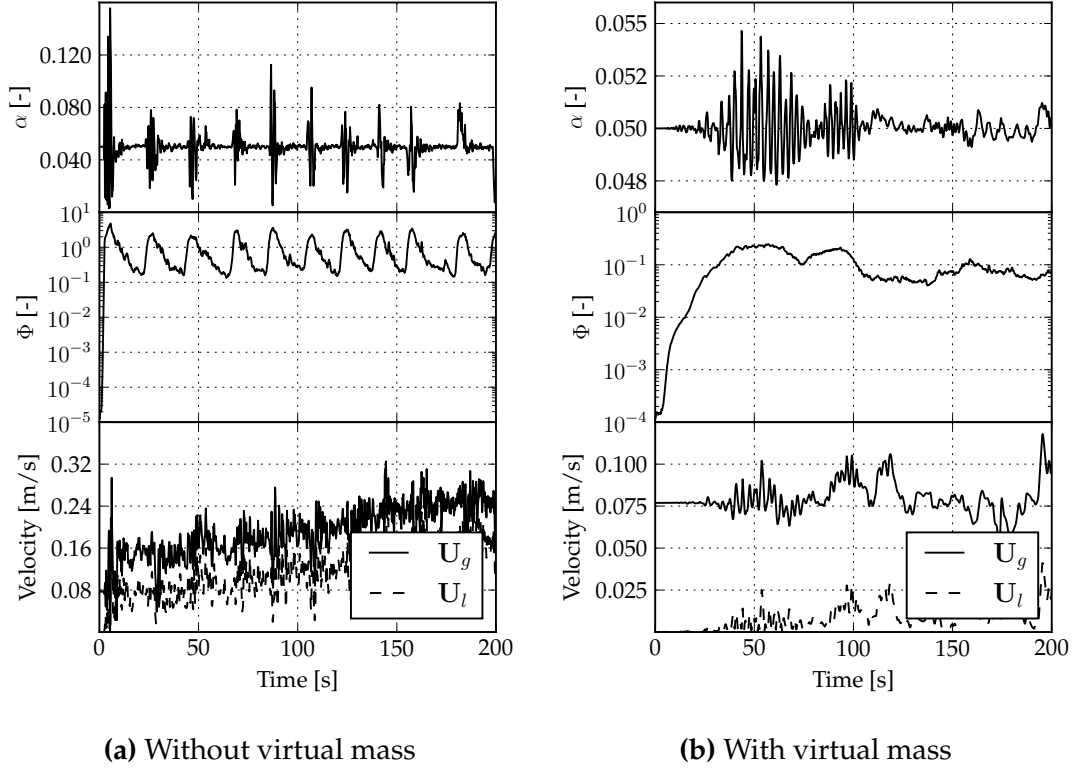
Liquid density, constant $\rho_l$	1000 kg/m <sup>3</sup>
Gas density, constant $\rho_g$	1 kg/m <sup>3</sup>
Liquid viscosity, constant $\mu_l$	10 <sup>-3</sup> Pa s
Gas viscosity, constant $\mu_g$	10 <sup>-3</sup> Pa s
Bubble size, constant $d_b$	0.68 mm
Void fraction, uniform initial condition $\alpha_g$	0.05
Liquid velocity, uniform initial condition	(0, 0, 0) m/s
Gas velocity, uniform initial condition	(0, 0, 0) m/s
Gauge pressure, uniform initial condition	0 Pa

The simulation is run for 200 s and snapshots of the void fraction distribution are presented for six chosen time steps in Figure 6.4. The figure presents a comparison between simulations performed with only the drag force (upper rows) or complemented with the virtual mass force (bottom rows). Considering first the results with only the drag force it is seen that the initially flat distribution of the void fraction has developed to a non-uniform state. For the results from  $t = 40$  s to  $t = 120$  s, a pattern with meso-scale structures is evident. However at  $t = 200$  s, a more disruptive void behavior is seen, with a checkerboard formation in the horizontal direction. In contrast, the results including the virtual mass force show initial strange patterns, perceived as being of a numerical character, whereas a more physically sound void fraction distribution is seen at the later time steps.

The results from Figure 6.4 are further examined in Figure 6.5, where the temporal development of the void fraction and the velocity field are presented to-



**Figure 6.4:** The gas fraction field displayed for 6 time steps (as indicated at the top of the figure) with the case of no virtual mass (top two rows) and virtual mass included (bottom two rows), displayed in the horizontal plane (rectangular figures) and the vertical plane (square figures). [Paper VII]



**Figure 6.5:** Temporal development of the void fraction field (top), the uniformity index (middle) and the magnitude of the velocity field for both phases (bottom). Both cases exhibit an initial transient in the void fraction and the uniformity index, although significantly faster for the case without the virtual mass force. The simulations are performed with the initial condition  $\alpha = 0.05$ . [Paper VII]

gether with the time-resolved uniformity index

$$\Phi(t) = \frac{\alpha_{g,\max} - \alpha_{g,\min}}{\alpha_{g,\text{ave}}}, \quad (6.8)$$

where  $\alpha_{g,\max}$  and  $\alpha_{g,\min}$  are the instantaneous maximum and minimum void fractions in the domain, respectively. The index is used as a measure of the global heterogeneity of the system. The average void fraction ( $\alpha_{g,\text{ave}}$ ) is directly given by the initial conditions for each of the simulations, and, for the presented results, a flat initial volumetric fraction of 5% gas is applied. The figure again shows a significant difference in the dynamics of the vapor fraction. For the cases with the drag only a repetitive instable behavior is visual. Whereas an initial rapid increase of  $\Phi$  is seen for the cases with the virtual mass, the later stage of the simulation follows a smoother behavior.

The results of the simulations are interesting from multiple perspectives. First, an initially homogeneous distribution of all fields quickly change and a variety of heterogeneous states is seen. The divergence from the uniform fields is interesting as there are no actual mechanisms in the equations forcing the change, and,

perhaps even more interesting as there are no mechanisms driving the system back to the smooth state. Second, the inclusion of the virtual mass force significantly changes the characteristics of the void fraction distribution. Although the actual magnitude of the force is small in comparison to the drag, the alteration of the behavior is distinct and seems to give a physically more plausible behavior. Third, it is interesting that the clean formulation, i.e. with no excessive diffusivity due to coarse turbulence modeling or additional momentum exchanges, exhibits clear instabilities. Such dynamic behavior would potentially be hidden with the inclusion of the mentioned type of terms, and arguably the latter would potentially hide the instabilities.

A key question raised in Paper VII is that of the trustworthiness of dynamic results from the two-fluid model. In detail, although the simulations might appear as sound and physical after the initial instabilities of numerical character (as was the case for the simulations including the virtual mass force), how are these results to be perceived? As argued in the paper, the quantitative values of the dynamic simulations are not immediately to be trusted which again emphasizes the complexities of the two-fluid model and the many questions marks yet to be resolved.



# Conclusions and recommendations for future work

Finally, the thesis is to be summarized and concluded. As the work was already split into two parts from the objectives point of view, the fine-mesh multiphysics and the two-phase flow studies, also the summary is presented in two parts. Last, I provide an outlook for the future of simulations for the coupled neutronic and thermal-hydraulic problem and how efforts like the one currently presented could be of importance for the nuclear industry in the future.

## 7.1 Fine-mesh multiphysics simulations

As the development of the multiphysics solver has been an integral part of the thesis, a summary of the methodology is first given. Second, a summary of the results is provided and accompanied by a conclusion on the achieved fidelity of the simulations.

### 7.1.1 Summary of the methodology

The presented methodology is aimed at fine-mesh simulations of the multiphysics problem of neutronics and thermal-hydraulics. The computational tool described in the thesis is based on the finite volume method and implemented in the open source framework OpenFOAM®. The thermal-hydraulic equations are solved by a CFD approach with segregated pressure and velocity solvers complemented by a RANS model for turbulence. The neutronic problem is handled by the multi-group diffusion equation (steady and transient simulations) and the discrete ordinates method (steady cases only), and solved with iterative, fixed-point, algorithms. The multiphysics couplings are handled in a Picard iteration style, with

sequential updates of each of the two modules. The multiphysics solver is parallelized based on the MPI implementation in OpenFOAM®, and with specific handling of the decomposition such that all overlapping cells of the meshes are kept at the same computational node.

The multiphysics solver is complemented by a utility for cross-section generation for sub-pin few group simulations. The tool is based on Serpent and handles a 2D fuel assembly geometry, specified in a configuration file. Similarly, a mesh tool is developed and to produce multi region body-fitted meshes for the thermal hydraulic problem, with resolved fuel, gap, cladding and moderator regions and a single, monolithic, mesh for the neutronics. The meshes are computed based on a block structure that gives a high level of user influence on the discretization.

### 7.1.2 Results and conclusions

The multiphysics tool was applied to both transient and steady simulations, where the results from the latter showed that:

- The fine-mesh simulations are able to provide the fuel and moderator temperature gradients on a sub-pin and resolved subchannel level with an equally high resolution neutronics solution on a quarter of a  $15 \times 15$  fuel assembly with PWR like thermophysical conditions.
- The convergence characteristic indicates that, generally, a few multiphysics iterations are required to resolve the couplings, whereas large number of sub-iterations are required for the neutronics and the thermal-hydraulics, respectively.
- The neutronics diffusion solver is, as expected, inferior to the  $S_N$  solver as shown on a two-dimensional case, validated against a Monte Carlo solution. The discrete ordinates method exhibits a significant ray effect which diminishes with the increasing order of the method.

As regards the transient simulations the results showed that:

- The fine-mesh simulation of a quarter of a  $7 \times 7$  fuel assembly captures the temporal development of the heterogeneities following a ramping transient at the inlet of the fuel assembly.
- The solver produces transient responses to local perturbations which could be verified with a novel approach based on the point-kinetic component of the system response.

In general, it can be concluded that all objectives were fulfilled for this part of the thesis. Nevertheless, there are also improvements to be developed (or implemented) for nearly every single aspect covered in the thesis and the solver methodology. To mention a few, it would be of interest and value to:

- Develop a transient version of the  $S_N$  solver. The steady-state validation results suggest that the diffusion approximation induces a significant error, not the least due to the highly resolved system and the resulting heterogeneous material regions. Clearly a transport method is required also for the transient cases. In addition, the  $S_N$  solver needs to be significantly accelerated, advisably based on a Krylov approach.
- Investigate the effect of a more detailed geometry, such as spacers. Such an effort would be even more interesting in combination with a LES approach to the turbulence to capture heterogeneities in the heat transfer and fluid thermo-physical state due to resolved fluctuations.
- Develop a multiphase version of the thermal-hydraulics method. As suggested in the introduction, the heterogeneities in the moderator are significantly larger for voided cases and thus a stronger fine-mesh coupling could be anticipated.
- Assess non-linear techniques to handle the couplings in the overall multi-physics problem as well as the separate modules. In particular, it would be interesting to implement the Anderson mixing methodology for the multi-physics problem and a more implicit approach to the pressure and velocity coupling.
- Add a multiscale methodology to compute boundary conditions for the fine-mesh simulations. Whereas the presented work was all based on periodic or symmetry boundary conditions in horizontal direction it would be of interest to investigate the effect of a more realistic environment of the simulated systems. To limit the computational effort a multiscale strategy would be a good candidate, simulating a hierarchy of scales in the same solver.

In addition, the implementation and the framework itself could be extended with other physics modules and with more generic simulation capabilities such as fluid-structure interaction or thermo-mechanical modules. Finally, it would be interesting to test the current approach for larger simulation domains, i.e. based on significantly larger computer resources.

## 7.2 Two-fluid simulations

The results and conclusions for the two-fluid simulations are separately reported for the DQMOM methodology and the dynamics investigations.

### 7.2.1 DQMOM coupled to a two-fluid solver

The presented methodology for subcooled boiling flows is based on a coupled PBE and two-fluid solver approach. The PBE is solved using DQMOM with a

newly proposed formulation for the condensation of bubbles. The method is compared to a MUSIG method, which in contrast to DQMOM applies a set of fixed bubble sizes.

The comparisons of the mentioned methodologies propose that:

- DQMOM needs significantly fewer discrete bubble sizes to reproduce the distribution as compared to the same accuracy as MUSIG. This further implies an edge in terms of shorter computational time required for DQMOM.
- DQMOM requires specific treatment for the extreme of very small bubbles, typical for subcooled flows where the bubbles grown at the wall fully condense in the bulk of the flow. Remedies in terms of regularization of the weights and the abscissas are shown for this purpose and successfully stabilize the solver.

In addition, the coupling to the two-fluid solver was carefully analyzed and, in particular, wall conditions for the insertion of bubbles were studied for both DQMOM and MUSIG.

As discussed in the analysis, the simulations presented in the thesis were not directly targeted to the BWR geometry or conditions. Instead, the study should be seen as an attempt to extend the field of CFD simulations for bubbly flows, and with particular focus on algorithms for the solution of PBEs. As a next step it would be of interest to investigate the benefits and drawbacks of the DQMOM method on a system more closely mimicking the subchannels in the reactor and of significantly larger size.

## 7.2.2 Two-fluid instability results

To investigate the dynamics of the two-fluid model, simulations based on adiabatic conditions were performed on fully periodic systems. The momentum exchange was based on the drag force and the virtual mass force only and no turbulence model was involved. The simulations showed that:

- The initially uniform void fraction distribution changed to a heterogeneous spatial distribution, exhibiting meso-scale structures.
- The inclusion of the virtual mass force stabilized the solver and resulted in a more physical character of the dynamics.

The analysis emphasized that the numerical character of the initial stage of the instabilities raises questions on the trustworthiness of such simulations and further illustrates the complexities of the two-fluid model. For future investigations it would be particularly interesting to evaluate the above proposed conclusions based on an entirely different methodology. In detail, to prove, or at least more firmly confirm, the existence of meso-scale structures, a comparison to a Lagrangian simulations framework should be a reasonable approach.

## 7.3 Future outlook

The core of a nuclear reactor is really an astonishing challenge from so many perspectives, and although high-fidelity multiphysics has been a hot topic in the last few years, much remains to be done.

For neutronics, the recent rapid development of the Monte Carlo solvers makes such types of methodologies important candidates for future tools. We have seen that there are many problematic aspects (not the least concerning the cross-section generation) which are completely eluded with the Monte Carlo approach and, although there are still gaps as regards transient simulations, this type of solver is likely to be an important component of future high-fidelity multiphysics tools.

As regards thermal-hydraulics, the areas of applications of CFD for reactor core simulations are likely to increase in number as well as in importance. The continuing growth of computational resources will successively allow for ever finer scales to be resolved and the application of high-resolution turbulence methods will for sure play an even more important role in future design of fuel assemblies and reactor cores. For multiphase CFD much theoretical work still remains, and, although, larger clusters can allow for more industrial use of interface tracking methodologies, full fine-mesh assembly simulations are for a long time still going to rely on averaged approaches, which are thus a continued important area for research.

Another future important question is that of validation of the novel multiphysics approaches. Although many of the suggested modules for the coupled tools can be separately validated, the community should aim at direct validation of the multiphysics solvers, in particular on fine-scales.



# Acknowledgements

The research performed in this thesis was financed by Svenskt Kärntekniskt Centrum (SKC) and the support is gratefully acknowledged. Additional grants for traveling were given by Åforsk, Kungliga Vetenskaps- och Vitterhets-Samhället (KVVS) and The Royal Swedish Academy of Science. The computations were performed on resources at Chalmers Centre for Computational Science and Engineering (C3SE) provided by the Swedish National Infrastructure for Computing (SNIC). Furthermore, the Swedish Research Council (Vetenskapsrådet) is acknowledged for the financing of the Development of Revolutionary and Accurate Methods for Safety Analyses of Future and Existing Reactors (DREAM4SAFER) framework grant (contract number C0467701), related to Papers III and VII in the thesis.

First, I would like to express my gratitudes to my supervisor Prof. Christophe Demaziere. Your long standing support and belief in all my ideas have been both an encouragement and privilege. I also thank my co-supervisors. Assoc. Prof. Paolo Vinai for your devotion to all my writing and your efforts on making them better. Prof. Srdjan Sasic, thank you for your guidance, encouragement and all insightful discussions on multiphase flow and more. Finally, I thank Prof. Imre Pazsit for being the examiner of the thesis.

I thank all colleagues I had the pleasure to work with during the past five years. Assistant Prof. Henrik Ström for interesting and fruitful work on the intricate two-fluid model. Sebastian Gonzalez-Pintor for insightful thoughts on neutronics and mathematics. Rasmus Andersson for the splendid Master thesis work. Assistant Prof. Gaetano Sardina for the DEM and two-fluid discussions. Prof. Håkan Nilsson for the opportunities of teaching and discussing OpenFOAM. Prof. Hrvoje Jasak for the inspiring talks on CFD and the opportunity to visit Zagreb. Vuko Vukcevic for inspiring discussions on coupled calculations. Ananda Subramani Kannan for the DEM discussions. Assoc. Prof. Julie Gold for the yearly doktorandsamtal.

Furthermore, I thank all my current and past colleagues at Nuclear Engineering for discussions, travel company, floorball and more. Thank you Daniel, Johan and Olov for the lunches. Thanks to all students I had the privilege to teach and

thanks to Assoc. Prof. Anders Nordlund for your support and encouragement in my teaching and the many interesting discussions. In addition, I thank my colleagues at Fraunhofer-Chalmers Research Centre for Industrial Mathematics for engaging work, inspiring discussions and motivation to get the PhD finished.

My warmest thanks to my family: my father for the almost daily encouraging discussions and all shared interests, my mother for the love and support, my sisters and their husbands for all enjoyable moments. Adam your support and discussions was more than once indispensable and your constant encouragement on the writing of this thesis was decisive.

Finally, thank you Noomi and Julia for the inspiration and help with getting priorities in life straight, and thank you Cornelia for your love, support and endless patience!



# Bibliography

- [1] ENDF/B-VII. Data extracted via EXFOR: Experimental Nuclear Reaction Data. URL: <https://www-nds.iaea.org/exfor/servlet/E4sGetTabSect?SectID=2292099&req=2031&PenSectID=7664758>.
- [2] J. Rhodes, K. Smith, D. Lee. "CASMO-5 development and applications". *Proc. ANS Topical Meeting on Reactor Physics (PHYSOR-2006)*. 2006, pp. 10–14.
- [3] D. Knott, A. Yamamoto. "Lattice Physics Computations". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [4] T. Bahadir, S.-Ö. Lindahl. "Studsviks next generation nodal code SIMULATE-5". *Advances in Nuclear Fuel Management IV (ANFM 2009)* (2009).
- [5] J. Hohorst, S. Polkinghorne, L. Siefken, C. Allison, C. Dobbe. *TMI-2 analysis using SCDAP/RELAP5/MOD3. 1*. Tech. rep. Lockheed Idaho Technologies Co., 1994.
- [6] A. Prosperetti, G. Tryggvason. *Computational Methods for Multiphase Flow*. Cambridge University Press, Cambridge, United Kingdom and New York, NY, USA, 2007.
- [7] R. K. Salko, M Avramova. "CTF theory manual". *The Pennsylvania State University* (2015).
- [8] M. Daeubler, A. Ivanov, B. L. Sjenitzer, V. Sanchez, R. Stieglitz, R. Macian-Juan. "High-fidelity coupled Monte Carlo neutron transport and thermal-hydraulic simulations using Serpent 2/SUBCHANFLOW". *Annals of Nuclear Energy* 83 (2015), pp. 352 –375.
- [9] C. Brennen. *Thermo-Hydraulics of Nuclear Reactors*. Dankat Publishing Company, 2013.
- [10] K. Ivanov, M. Avramova. "Challenges in coupled thermal-hydraulics and neutronics simulations for LWR safety analysis". *Annals of Nuclear Energy* 34.6 (2007), pp. 501–513.
- [11] IAEA. *Best estimate safety analysis for nuclear power plants: Uncertainty evaluation*. Tech. rep. IAEA, Vienna, 2008.
- [12] U. Imke, V. H. Sanchez. "Validation of the subchannel code SUBCHANFLOW using the NUPEC PWR tests (PSBT)". *Science and Technology of Nuclear Installations* (2012).

- [13] D. Bestion. "From the Direct Numerical Simulation to System Codes-Perspective for the Multi-scale Analysis of LWR Thermalhydraulics". *Nuclear Engineering and Technology* 42.6 (2010), pp. 608–619.
- [14] J. C. Ragusa, V. S. Mahadevan. "Consistent and accurate schemes for coupled neutronics thermal-hydraulics reactor analysis". *Nuclear Engineering and Design* 239.3 (2009), pp. 566–579.
- [15] M. Ellis, J. Watson, K. Ivanov. "Progress in the development of an implicit steady state solution in the coupled code TRACE/PARCS". *Progress in Nuclear Energy* 66 (2013), pp. 1–12.
- [16] M. E. Conner, E. Baglietto, A. M. Elmahdi. "CFD methodology and validation for single-phase flow in PWR fuel assemblies". *Nuclear Engineering and Design* 240.9 (2010), pp. 2088–2095.
- [17] F. Jatuff, F. Giust, J. Krouthen, S. Helmersson, R. Chawla. "Effects of void uncertainties on the void reactivity coefficient and pin power distributions for a 10x10 BWR assembly". *Ann. Nucl. Energy* 33 (2006), pp. 119–125.
- [18] T. Ikehara, Y. Kudo, M. Tamitani, M. Yamamoto. "Effect of Subchannel Void Fraction Distribution on Lattice Physics Parameters for Boiling Water Reactor Fuel Bundles". *J. Nucl. Sci. Technol.* 45.12 (2008), pp. 1237–1251.
- [19] D. R. Gaston. "Physics-based multiscale coupling for full core nuclear reactor simulation". *Annals of Nuclear Energy* 84 (2015), pp. 45–54.
- [20] V. Petrov, B. K. Kendrick, D. Walter, A. Manera, J. Secker. "Prediction of CRUD deposition on PWR fuel using a state-of-the-art CFD-based multi-physics computational tool". *Nuclear Engineering and Design* 299 (2016), pp. 95–104.
- [21] CASL. *Successful expanded prediction of the nature of CRUD found in pressurized water reactor coolant*. (last visited June 19, 2017). URL: <http://www.casl.gov/highlights/crud.shtml>.
- [22] J. Bakosi, M. A. Christon, R. B. Lowrie, L. Pritchett-Sheats, R. Nourgaliev. "Large-eddy simulations of turbulent flow for grid-to-rod fretting in nuclear reactors". *Nuclear Engineering and Design* 262 (2013), pp. 544–561.
- [23] U. Bieder, F. Falk, G. Fauchet. "LES analysis of the flow in a simplified PWR assembly with mixing grid". *Progress in Nuclear Energy* 75 (2014), pp. 15–24.
- [24] E. Merzari, A. Obabko, P. Fischer, N. Halford, J. Walker, A. Siegel, Y. Yu. "Large-scale large eddy simulation of nuclear reactor flows: Issues and perspectives". *Nuclear Engineering and Design* 312 (2017), pp. 86–98.
- [25] J. Cardoni, Rizwan-uddin. "Nuclear Reactor Multi-Physics Simulations with Coupled MCNP5 and STAR-CCM+". *M&C 2011, Rio de Janeiro, Brazil* (2011).
- [26] L. Li, K. Wang. "The first-principle coupled calculations using TMCC and CFX for the pin-wise simulation of LWR". *Proc. Int. Conf. PHYSOR, American Nuclear Society, Knoxville, Tennessee, USA*. 2012.
- [27] A. Ivanov, V. Sanchez, R. Stieglitz, K. Ivanov. "High fidelity simulation of conventional and innovative LWR with the coupled Monte-Carlo thermal-hydraulic system MCNP-SUBCHANFLOW". *Nuclear Engineering and Design* 262 (2013), pp. 264–275.

- 
- [28] D. P. Weber. "High-Fidelity Light Water Reactor Analysis with the Numerical Nuclear Reactor". *Nucl. Sci. Eng.* 155.155 (2006), pp. 395–408.
- [29] J. Yan, B. Kochunas, M. Hursin, T. Downar, Z. Karoutas, E. Baglietto. "Coupled Computational Fluid Dynamics and MOC Neutronic Simulations of Westinghouse PWR Fuel Assemblies with Grid Spacers". *NURETH-14, Toronto, Ontario, Canada, September 25-30* (2011).
- [30] B. Kochunas, S. Stimpson, B. Collins, T. Downar, R. Brewster, E. Baglietto, J. Yan. "Coupled Full Core Neutron Transport/CFD Simulations of Pressurized Water Reactors". *PHYSOR 2012, Knoxville, Tennessee, USA, April 15-20* (2012).
- [31] R. Schmidt. "An approach for coupled-code multiphysics core simulations from a common input". *Annals of Nuclear Energy* 84 (2015), pp. 140–152.
- [32] D. Gaston, C. Newman, G. Hansen, D. Lebrun-Grandie. "MOOSE: A parallel computational framework for coupled systems of nonlinear equations". *Nuclear Engineering and Design* 239.10 (2009), pp. 1768–1778.
- [33] J. Bell, B. Lewis. "CANDU fuel bundle deformation modelling with COMSOL multiphysics". *Nuclear Engineering and Design* 250 (2012), pp. 134–141.
- [34] R. Liu, A. Prudil, W. Zhou, P. K. Chan. "Multiphysics coupled modeling of light water reactor fuel performance". *Progress in Nuclear Energy* 91 (2016), pp. 38–48.
- [35] C. Fiorina, I. Clifford, M. Aufiero, K. Mikityuk. "GeN-Foam: a novel OpenFOAM® based multi-physics solver for 2D/3D transient analysis of nuclear reactors". *Nuclear Engineering and Design* 294 (2015), pp. 24–37.
- [36] C. Newman, G. Hansen, D. Gaston. "Three dimensional coupled simulation of thermomechanics, heat, and oxygen diffusion in UO<sub>2</sub> nuclear fuel rods". *Journal of Nuclear Materials* 392.1 (2009), pp. 6–15.
- [37] J. A. Turner, K. Clarno, M. Sieger, R. Bartlett, B. Collins, R. Pawlowski, R. Schmidt, R. Summers. "The Virtual Environment for Reactor Applications (VERA): Design and architecture". *Journal of Computational Physics* 326 (2016), pp. 544–568.
- [38] M. A. Christon, R. Lu, J. Bakosi, B. T. Nadiga, Z. Karoutas, M. Berndt. "Large-eddy simulation, fuel rod vibration and grid-to-rod fretting in pressurized water reactors". *Journal of Computational Physics* 322 (2016), pp. 142–161.
- [39] J. R. Lee, J. Kim, C.-H. Song. "Synthesis of the turbulent mixing in a rod bundle with vaned spacer grids based on the OECD-KAERI CFD benchmark exercise". *Nuclear Engineering and Design* 279 (2014), pp. 3–18.
- [40] S. Lo, J. Osman. "CFD modeling of boiling flow in PSBT 5 5 bundle". *Science and Technology of Nuclear Installations* (2012).
- [41] T. Sofu, J. Thomas, D. Weber, W. Pointer. "Coupled BWR Calculations with the Numerical Nuclear Reactor Software System". *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*. 2007.
- [42] D. Bestion. "The difficult challenge of a two-phase {CFD} modelling for all flow regimes". *Nuclear Engineering and Design* 279 (2014). {SI} : CFD4NRS-4, pp. 116–125.
- [43] *Univac LARC PROGRAMMING: The Computing Unit*. Remington Rand Univac. 1961.

- [44] CDC6600. (last visited June 19, 2017). Wikipedia. URL: [https://en.wikipedia.org/wiki/CDC\\_6600](https://en.wikipedia.org/wiki/CDC_6600).
- [45] D. Loshin. "Chapter 2 - History". *High Performance Computing Demystified*. Ed. by David Loshin. Academic Press, 1994, pp. 13–21.
- [46] Blue Gene. (last visited June 19, 2017). Wikipedia. URL: [https://en.wikipedia.org/wiki/Blue\\_Gene](https://en.wikipedia.org/wiki/Blue_Gene).
- [47] P. J. Turinsky. "Advances in multi-physics and high performance computing in support of nuclear reactor power systems modeling and simulation". *Nuclear Engineering and Technology* 44.2 (2012), pp. 103–122.
- [48] H. Sutter. *The Free Lunch Is Over: A Fundamental Turn Toward Concurrency in Software*. (last visited March 3, 2017). Dr. Dobbs's Journal, 2005. URL: <http://www.gotw.ca/publications/concurrency-ddj.htm>.
- [49] A. Danowitz, K. Kelley, J. Mao, J. P. Stevenson, M. Horowitz. "CPU DB: Recording Microprocessor History". *Commun. ACM* 55.4 (Apr. 2012), pp. 55–63.
- [50] C. DB. (last visited June 19, 2017). Stanford VLSI Group. URL: <http://cpudb.stanford.edu/processors>.
- [51] T. S. Sites. *Performance Development*. (last visited June 19, 2017). URL: <https://www.top500.org/statistics/perfdevel/>.
- [52] Intel Math Kernel Library Benchmarks. Linux package l\_mklb\_p\_2017.2.015. URL: <https://software.intel.com/en-us/articles/intel-mkl-benchmarks-suite>.
- [53] J. Demmel, M. Hoemmen, M. Mohiyuddin, K. Yelick. "Avoiding communication in sparse matrix computations". *Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on*. IEEE. 2008, pp. 1–12.
- [54] C. Lomont. *Introduction to Intel Advanced Vector Extensions*, June 2011.
- [55] K. Yuhin. *Intel Advanced Vector Extensions 2015/2016 Support in GNU Compiler Collection. GNU Tools Cauldron 2014*. (last visited June 19, 2017). Intel. 2014. URL: [https://gcc.gnu.org/wiki/cauldron2014?action=AttachFile&do=get&target=Cauldron14\\_AVX-512\\_Vector\\_ISA\\_Kirill\\_Yuhin\\_20140711.pdf](https://gcc.gnu.org/wiki/cauldron2014?action=AttachFile&do=get&target=Cauldron14_AVX-512_Vector_ISA_Kirill_Yuhin_20140711.pdf) (visited on 04/21/2017).
- [56] N. CUDA. *CUDA Zone*. 2017.
- [57] C. Calvin, D. Nowak. "High Performance Computing in Nuclear Engineering". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [58] MPI Forum. *MPI Forum*. (last visited June 19, 2017). URL: <http://mpi-forum.org/>.
- [59] SIEMENS MDX. *STAR-CCM+*. (last visited June 19, 2017). URL: <http://mdx.plm.automation.siemens.com/star-ccm-plus>.
- [60] ANSYS. *ANSYS Fluent: CFD Simulation*. (last visited June 19, 2017). URL: <http://www.ansys.com/Products/Fluids/ANSYS-Fluent>.
- [61] Gerris. *Gerris Flow Solver*. (last visited June 19, 2017). Apr. 13, 2017. URL: <http://gfs.sourceforge.net/>.

- 
- [62] OpenFOAM Foundation. *The OpenFOAM Foundation*. (last visited June 19, 2017). URL: <https://openfoam.org/>.
- [63] Multiphase Flow Sciences Group at NETL. *MFIX*. (last visited June 19, 2017). URL: <https://mfix.netl.doe.gov/>.
- [64] F. Harlow, J. Fromm. "Dynamics and heat transfer in the von Krmn wake of a rectangular cylinder". *Physics of Fluids* 7.8 (1964), pp. 1147–1156.
- [65] J. H. Ferziger, M. Peric. *Computational methods for fluid dynamics*. 1997.
- [66] COMSOL. *COMSOL CFD Module*. (last visited June 19, 2017). 2017. URL: <https://www.comsol.se/cfd-module>.
- [67] E. Dick. "Introduction to Finite Element Methods in Computational Fluid Dynamics". *Computational Fluid Dynamics*. Ed. by John F. Wendt. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 235–274.
- [68] A. Hébert. *Multigroup Neutron Transport and Diffusion Computations*. Springer-Verlag, 2010, pp. 751–911.
- [69] S. Patankar. *Numerical heat transfer and fluid flow*. CRC press, 1980.
- [70] H Jasak. "Error analysis and estimation in the finite volume method with applications to 1027 fluid flows". PhD thesis. PhD thesis, Imperial College, University of London. 1028, 1996.
- [71] F. Bassi, S. Rebay. "High-order accurate discontinuous finite element solution of the 2D Euler equations". *Journal of computational physics* 138.2 (1997), pp. 251–285.
- [72] Y. Saad, H. A. Van Der Vorst. "Iterative solution of linear systems in the 20th century". *Journal of Computational and Applied Mathematics* 123.1 (2000), pp. 1–33.
- [73] Y. Saad. *Iterative methods for sparse linear systems*. SIAM, 2003.
- [74] Y. Shapira. *Matrix-based multigrid: theory and applications*. Vol. 2. Springer Science & Business Media, 2008.
- [75] H. G. Weller, G Tabor, H. Jasak, C Fureby. "A tensorial approach to computational continuum mechanics using object-oriented techniques". *Computers in physics* 12.6 (1998), pp. 620–631.
- [76] Wikki. *foam-extend-3.1*. (last visited February 16, 2015). 2015. URL: <http://wikki.gridcore.se/foam-extend/foam-extend-3-1-zagreb>.
- [77] OpenFOAM. *OpenFOAM User's Guide*. *User's Guide*. Version 2.1.1. OpenCFD Limited. OpenFOAM Foundation, 2012.
- [78] OpenFOAM. *OpenFOAM Programmer's Guide*. *Programmers Guide*. Version 2.0.0. OpenCFD Limited. OpenFOAM Foundation, 2011.
- [79] T. Maric, J. Höpken, K. Mooney. *The OpenFOAM Technology Primer*. Stan Mott, 2014.
- [80] F Moukalled, L Mangani, M Darwish. *The finite volume method in computational fluid dynamics*. Springer, 2016.
- [81] S. Patankar, D. Spalding. "A calculation procedure for heat, mass and momentum transfer in three dimensional parabolic flows". *International Journal of Heat and Mass Transfer* 15 (1972), pp. 1787–1806.

- [82] R. Issa, B. Ahmadi-Befrui, K. Beshay, A. Gosman. "Solution of the implicitly discretised reacting flow equations by operator-splitting". *Journal of Computational Physics* 93.2 (1991), pp. 388–410.
- [83] C. Rhie, W. Chow. "A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation". *AIAA Journal* 21 (1983), pp. 1525–1532.
- [84] J. Ahrens, B. Geveci, C. Law, C. Hansen, C. Johnson. "36-ParaView: An End-User Tool for Large-Data Visualization". *The Visualization Handbook* (2005), p. 717.
- [85] F. Juretić, A. Gosman. "Error analysis of the finite-volume method with respect to mesh type". *Numerical heat transfer, part B: fundamentals* 57.6 (2010), pp. 414–439.
- [86] H. Jasak, A. Gosman. "Automatic resolution control for the finite-volume method, Part 2: Adaptive mesh refinement and coarsening". *Numerical Heat Transfer: Part B: Fundamentals* 38.3 (2000), pp. 257–271.
- [87] G. Hansen, S. Owen. "Mesh generation technology for nuclear reactor simulation; barriers and opportunities". *Nuclear Engineering and Design* 238.10 (2008), pp. 2590–2605.
- [88] T. J. Tautges, R. Jain. "Creating geometry and mesh models for nuclear reactor core geometries using a lattice hierarchy-based approach". *Engineering with Computers* 28.4 (Oct. 1, 2012), p. 319.
- [89] K. Podila, Y. Rao. "{CFD} modelling of turbulent flows through 5 × 5 fuel rod bundles with spacer-grids". *Annals of Nuclear Energy* 97 (2016), pp. 86–95.
- [90] SALOME Platform. 2017. URL: <http://www.salome-platform.org/>.
- [91] C. Chauliac, J.-M. Aragonés, D. Bestion, D. G. Cacuci, N. Crouzet, F.-P. Weiss, M. A. Zimmermann. "NURESIM—A European simulation platform for nuclear reactor safety: Multi-scale and multi-physics calculations, sensitivity and uncertainty analysis". *Nuclear Engineering and Design* 241.9 (2011), pp. 3416–3426.
- [92] A. Rubin, A. Schoedel, M. Avramova, H. Utsuno, S. Bajorek, A. Velazquez-Lozada. "OECD/NRC Benchmark based on NUPEC PWR subchannel and bundle tests (PSBT), Volume I: Experimental Database and Final Problem Specifications". *US NRC OECD Nuclear Energy Agency* (2010).
- [93] ANSYS® ICEM. *ANSYS ICEM CFD Tutorial Manual*. 2013.
- [94] D. Lebrun-Grandie, J. C. Ragusa, R. Sampath. "Efficient finite element field interpolation for multiphysics applications". *PHYSOR 2014, September 28 - October 3, Japan* (2014).
- [95] Studsvik. *CASMO 4e Manual*. 2009.
- [96] G. Marleau, A. Hebert, R. Roy. *A User Guide for DRAGON*. 2013.
- [97] Los Alamos National Laboratory. *MCNP*. (last visited June 19, 2017). Apr. 20, 2017. URL: <https://mcnp.lanl.gov/>.
- [98] J. E. Hoogenboom, V. A. Khotylev, J. M. Tholammakkil. "Generation of multi-group cross sections and scattering matrices with the Monte Carlo code MCNP5". *Joint International Topical Meeting on Mathematics & Computation and Supercomputing in Nuclear Applications*. 2007, pp. 15–19.

- 
- [99] J. Kuijper, S. Van Der Marck, A Hogenbirk. "Using homogenized macroscopic group cross sections in continuous-energy Monte Carlo neutron transport calculations with MCNP". *proc. M&C+ SNA* (2007), pp. 15–19.
  - [100] E. Pettersen, C. Demazière, K. Jareteg, E. Schönfeldt T, B. Lauritzen. "Development of a Monte-Carlo based method for calculating the effect of stationary fluctuations". *M&C 2015, Nashville, Tennessee* (2015).
  - [101] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta, T. Kaltiaisenaho. "The Serpent Monte Carlo code: Status, development and applications in 2013". *Annals of Nuclear Energy* 82 (2015), pp. 142–150.
  - [102] C. Demazière. "Investigation of the bias coming from spectrum corrections in the simulations of nuclear reactor transients". Vol. 6. 2016, pp. 3561–3570.
  - [103] M. Aufiero, C. Fiorina, A. Laureau, P. Rubiolo, V. Valtavirta. "Serpent–OpenFOAM coupling in transient mode: simulation of a Godiva prompt critical burst". *Proceedings of M&C+ SNA+ MC* (2015), pp. 19–23.
  - [104] V Valtavirta, T Ikonen, T Viitanen, J Leppänen. *Simulating fast transients with fuel behavior feedback using the Serpent 2 Monte Carlo code*. Tech. rep. 2015.
  - [105] B. L. Sjenitzer, J. E. Hoogenboom. "A Monte Carlo method for calculation on the dynamic behaviour of nuclear reactors". *Progress in Nuclear Science and Technology* 2 (2011), pp. 716–721.
  - [106] A. K. Prinja, E. W. Larsen. "General Principles of Neutron Transport". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010, pp. 427–542.
  - [107] G. I. Bell, S. Glasstone. *Nuclear reactor theory*. Vol. 252. Van Nostrand Reinhold New York, 1970.
  - [108] E. W. Larsen, J. E. Morel. "Advances in discrete-ordinates methodology". *Nuclear Computational Science*. Springer, 2010, pp. 1–84.
  - [109] C. Demazière. *Modelling of nuclear reactors*. Division of Nuclear Engineering, Department of Applied Physics, Chalmers University of Technology, 2011.
  - [110] I. K. Abu-Shumays. "Angular Quadrature for Improved Transport Computations". *Transport Theory and Statistical Physics* 30.2-3 (2001), pp. 169–204.
  - [111] M. L. Adams, E. W. Larsen. "Fast iterative methods for discrete-ordinates particle transport calculations". *Progress in nuclear energy* 40.1 (2002), pp. 3–159.
  - [112] E. W. Larsen. "Diffusion-synthetic acceleration methods for discrete-ordinates problems". *Transport Theory and Statistical Physics* 13.1-2 (1984), pp. 107–126.
  - [113] J. S. Warsa, T. A. Wareing, J. E. Morel. "Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional  $S_n$  calculations in problems with material discontinuities". *Nuclear science and engineering* 147.3 (2004), pp. 218–248.
  - [114] R. Panton. *Incompressible flow*. Third edition. Wiley, 2005.
  - [115] J. C. Tannehill, D. A. Anderson, R. H. Pletcher. "Computational fluid dynamics and heat transfer". *Series in Computational and Physical Processes in Mechanics and Thermal Sciences* (1997).
  - [116] S. B. Pope. *Turbulent flows*. 2001.

- [117] S. Benhamadouche. "On the use of (U)RANS and {LES} approaches for turbulent incompressible single phase flows in nuclear engineering applications". *Nuclear Engineering and Design* 312 (2017). 16th International Topical Meeting on Nuclear Reactor Thermal Hydraulics, pp. 2–11.
- [118] H. K. Versteeg, W. Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [119] L. Davidson. "Fluid mechanics, turbulent flow and turbulence modeling". *Chalmers University of Technology, Goteborg, Sweden (Mar 2017)* (2017).
- [120] H. Elman. "Preconditioning Strategies for Models of Incompressible Flow". English. *Journal of Scientific Computing* 25.1 (2005), pp. 347–366.
- [121] E. Brakkee, P. Wilders. *A domain decomposition method for the advection-diffusion equation*. Tech. rep. Technische Universiteit Delft, 1994.
- [122] P. Lions. "On the Schwarz alternating method. III: a variant for non-overlapping domains". *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, Houston, Texas, March 20-22, (1990)*.
- [123] M. Modest. *Radiative Heat Transfer*. Academic Press, San Diego, USA, 2003.
- [124] K. Jareteg. *Development of an integrated deterministic neutronic/thermal-hydraulic model using a CFD solver*. M.Sc. thesis, Chalmers University of Technology. 2012.
- [125] M. Berrill, K. Clarno, S. Hamilton, R. Pawlowski. *Evaluation of Coupling Approaches*. Tech. rep. CASL-U-2014-0081-000, Oak Ridge National Laboratory, 2014.
- [126] D. A. Knoll, D. E. Keyes. "Jacobian-free Newton–Krylov methods: a survey of approaches and applications". *Journal of Computational Physics* 193.2 (2004), pp. 357–397.
- [127] M. A. Pope, V. A. Mousseau. "Accuracy and efficiency of a coupled neutronics and thermal hydraulics model". *Nuclear Engineering and Technology* 41.7 (2009), pp. 885–892.
- [128] D. G. Anderson. "Iterative procedures for nonlinear integral equations". *Journal of the ACM (JACM)* 12.4 (1965), pp. 547–560.
- [129] A. Toth, C. Kelley, S. Slattery, S. Hamilton, K. Clarno, R. Pawlowski. "Analysis of Anderson acceleration on a simplified neutronics/thermal hydraulics system". *Proceedings of MC2015, Nashville, USA* (2015).
- [130] M. T. Calef, E. D. Fichtl, J. S. Warsa, M. Berndt, N. N. Carlson. "Nonlinear Krylov acceleration applied to a discrete ordinates formulation of the k-eigenvalue problem". *Journal of Computational Physics* 238 (2013), pp. 188–209.
- [131] J. Willert, H. Park, W. Taitano. "Using Anderson acceleration to accelerate the convergence of neutron transport calculations with anisotropic scattering". *Nuclear Science and Engineering* 181.3 (2015), pp. 342–350.
- [132] W. Liu, L. Zhang, Y. Zhong, Y. Wang, Y. Che, C. Xu, X. Cheng. "Cfd high-order accurate scheme jacobian-free newton krylov method". *Computers & Fluids* 110 (2015), pp. 43–47.
- [133] S. Schunert. "A flexible nonlinear diffusion acceleration method for the S N transport equations discretized with discontinuous finite elements". *Journal of Computational Physics* 338 (2017), pp. 107–136.



- 
- [134] G. Kim, S. Kim, Y. Kim. "Parallelized Unstructured-Grid Finite Volume Method for Modeling Radiative Heat Transfer". *Journal of Mechanical Science and Technology* 19 (2005), pp. 1006–1017.
- [135] S. Plimpton, B. Hendrickson, S. Burns, W. McLendon, L. Rauchwerger. "Parallel Sn Sweeps on Unstructured Grids: Algorithms for Priorization, Grid Partitioning and Cycle Detection". *Nuclear Science and Engineering* 150 (2005), pp. 267–283.
- [136] Y. Saad. *Numerical Methods for Large Eigenvalue Problems: Revised Edition*. SIAM, 2011.
- [137] R. Stammler, M. Abbate. *Methods of steady state reactor physics in nuclear design*. Academic Press, London, 1983.
- [138] E. Lewis, W. J. Miller. *Computational Methods of Neutron Transport*. John Wiley & Sons, Inc., USA, 1984.
- [139] M. DeHart. "A discrete ordinates approximation to the neutron transport equation applied to generalized geometries". PhD thesis. Texas A&M University, 1992.
- [140] D Ginestar, G Verdú, V Vidal, R Bru, J Marín, J. Munoz-Cobo. "High order backward discretization of the neutron diffusion equation". *Annals of Nuclear Energy* 25.1-3 (1998), pp. 47–64.
- [141] N. Todreas, M. Kazimi. *Nuclear systems I: Thermal hydraulic fundamentals*. Taylor & Francis, Levittown, USA, 1993.
- [142] R. Nigmatulin. "Spatial averaging in the mechanics of heterogeneous and dispersed systems". *International Journal of Multiphase Flow* 5.5 (1979), pp. 353–385.
- [143] R. Lahey, D. Drew. "The three-dimensional time and volume averaged conservation equations of two-phase flow". *Advances in Nuclear Science & Technology* (1989).
- [144] G Yadigaroglu. "CMFD and the critical-heat-flux grand challenge in nuclear thermal-hydraulics—A letter to the Editor of this special issue". *International Journal of Multiphase Flow* 67 (2014), pp. 3–12.
- [145] Y. Sato, B. Niceno. "A depletable micro-layer model for nucleate pool boiling". *Journal of Computational physics* 300 (2015), pp. 20–52.
- [146] G Yadigaroglu. "CMFD (a brand name) and other acronyms". *International Journal of Multiphase Flow* 29.4 (2003), pp. 719–720.
- [147] G Tryggvason, J Lu. "DNS for multiphase flow model generation and validation". *NURETH-14, Toronto, Ontario, Canada, September 25-30 (2011)*.
- [148] N. Deen, J. Kuipers. "Direct Numerical Simulation (DNS) of mass, momentum and heat transfer in dense fluid-particle systems". *Current Opinion in Chemical Engineering* 5 (2014), pp. 84–89.
- [149] W. F. Noh, P. Woodward. "SLIC (simple line interface calculation)". *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*. Springer. 1976, pp. 330–340.
- [150] C. W. Hirt, B. D. Nichols. "Volume of fluid (VOF) method for the dynamics of free boundaries". *Journal of Computational Physics* 39.1 (1981), pp. 201–225.

- [151] S. Osher, J. A. Sethian. "Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations". *Journal of Computational Physics* 79.1 (1988), pp. 12–49.
- [152] M. Sussman, P. Smereka, S. Osher. "A level set approach for computing solutions to incompressible two-phase flow". *Journal of Computational physics* 114.1 (1994), pp. 146–159.
- [153] S. Unverdi, G. Tryggvason. "A front-tracking method for viscous, incompressible, multi-fluid flows". *Journal of computational physics* 100.1 (1992), pp. 25–37.
- [154] Q. Zeng, J. Cai, H. Yin, X. Yang, T. Watanabe. "Numerical simulation of single bubble condensation in subcooled flow using OpenFOAM". *Progress in Nuclear Energy* 83 (2015), pp. 336–346.
- [155] N. Samkhaniani, M. Ansari. "Numerical simulation of bubble condensation using CF-VOF". *Progress in Nuclear Energy* 89 (2016), pp. 120 –131.
- [156] M. Ishii. "Two-fluid model for two-phase flow". *Multiphase science and technology* 5.1-4 (1990).
- [157] M. Ishii, T. Hibiki. *Thermo-fluid dynamics of two-phase flow*. Springer Science & Business Media, 2010.
- [158] A. Prosperetti. "Ensemble Averaging Techniques for Disperse Flows". *Particulate Flows*. Vol. 98. The IMA Volumes in Mathematics and its Applications. Springer New York, 1998, pp. 99–136.
- [159] H. Weller. *Derivation, Modeling and Solution of the Conditionally Averaged Two-Phase Flow Equations*. Tech. rep. OpenCFD, 2005.
- [160] H. Rusche. "Computational Fluid Dynamics of Dispersed Two-Phase Flows at High Phase Fractions". PhD thesis. Imperial College of Science, Technology & Medicine, 2002.
- [161] D. Lhuillier, T. Theofanous, M.-S. Liou. "Multiphase Flows: Compressible Multi-Hydrodynamics". *Handbook of Nuclear Engineering*. Ed. by Dan Gabriel Cacuci. Springer US, 2010.
- [162] T. Dinh, R. Nourgaliev, T. Theofanous. "Understanding the ill-posed two-fluid model". *Proceedings of the 10th International topical meeting on nuclear reactor thermal-hydraulics (NURETH03)*. 2003.
- [163] H. A. Jakobsen. "Chemical Reactor Modeling". *Multiphase Reactive Flows, Berlin, Germany: Springer-Verlag* (2008).
- [164] G. Espinosa-Paredes. "A Derivation of the Nonlocal Volume-Averaged Equations for Two-Phase Flow Transport". *Science and Technology of Nuclear Installations* 2012 (2012).
- [165] S. Chu, A. Prosperetti. "On flux terms in volume averaging". *International Journal of Multiphase Flow* 80 (), pp. 176 –180.
- [166] M. Z. Podowski, R. M. Podowski. "Mechanistic multidimensional modeling of forced convection boiling heat transfer". *Science and Technology of Nuclear Installations* 2009 (2008).

- 
- [167] H. Anglart, O. Nylund. "CFD application to prediction of void distribution in two-phase bubbly flows in rod bundles". *Nuclear Engineering and Design* 163.1-2 (1996), pp. 81–98.
- [168] T. Hibiki, M. Ishii. "Interfacial area concentration of bubbly flow systems". *Chemical Engineering Science* 57.18 (2002), pp. 3967–3977.
- [169] E. Krepper, D. Lucas, T. Frank, H.-M. Prasser, P. J. Zwart. "The inhomogeneous MUSIG model for the simulation of polydispersed flows". *Nuclear Engineering and Design* 238.7 (2008), pp. 1690 –1702.
- [170] B. Selma, R. Bannari, P. Proulx. "Simulation of bubbly flows: Comparison between direct quadrature method of moments (DQMOM) and method of classes (CM)". *Chemical Engineering Science* 65.6 (2010), pp. 1925–1941.
- [171] G. Yeoh, C. Sherman, J. Tu. "On the prediction of the phase distribution of bubbly flow in a horizontal pipe". *Chemical Engineering Research and Design* 90 (2012), pp. 40–51.
- [172] E. Krepper, B. Koncar, Y. Egorov. "CFD modelling of subcooled boiling - Concept, validation and application to fuel assembly design". *Nuclear Engineering and Design* 237 (2007), pp. 716–731.
- [173] E. Krepper, R. Rzehak, C. Lifante, T. Frank. "CFD for subcooled flow boiling: Coupling wall boiling and population balance models". *Nuclear Engineering and Design* 255 (2013), pp. 330–346.
- [174] R. Rzehak, E. Krepper. "CFD for Subcooled Flow Boiling: Parametric Variations". *Science and Technology of Nuclear Installations* (2013).
- [175] D. Marchisio, R. Fox. "Solution of population balance equations using the direct quadrature method of moments". *Aerosol Science* 36 (2005), pp. 43–73.
- [176] D. Marchisio, A. Barresi, G. Baldi, R. Fox. "Comparison between the Classes Method and the Quadrature Method of Moments for multiphase systems". *8th conference "Multiphase flow in industrial plants", Alba, Italy* (2002).
- [177] R. Fox, F. Laurent, M. Massot. "Numerical simulation of spray coalescence in an Eulerian framework: direct quadrature method of moments and multi-fluid method". *Journal of Computational Physics* 227 (2008), pp. 3058–3088.
- [178] D. Ramkrishna. *Population balances: Theory and applications to particulate systems in engineering*. Academic Press, 2000.
- [179] S. A. Issa, P. Weisensee, R. Macin-Juan. "Experimental investigation of steam bubble condensation in vertical large diameter geometry under atmospheric pressure and different flow conditions". *International Journal of Heat and Mass Transfer* 70.0 (2014), pp. 918 –929.
- [180] R. Mudde, W. Harteveld, H. Akker. "Uniform flow in bubble columns". *Industrial & Engineering Chemistry Research* 48.1 (2008), pp. 148–158.
- [181] K. Agrawal, P. Loezos, M. Syamlal, S. Sundaresan. "The role of meso-scale structures in rapid gas-solid flows". *Journal of Fluid Mechanics* 445 (2001), pp. 151–185.
- [182] K. Jareteg, H. Ström, S. Sasic, C. Demazière. "Numerical investigation of instabilities in the two-fluid model for CFD simulations of LWRs". *M&C 2015, Nashville, Tennessee* (2015).